Adaptive Networks for Robotics and the emergence of Reward Anticipatory Circuits

Gary McHale

School of Engineering and Informatics, Department of Informatics

University of Sussex

A thesis submitted for the degree of

D.Phil in Computer Science and Artificial Intelligence

September 30th, 2011

DECLARATION

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for a degree.

Signature:

Dedicated to my son Taro, who I hope one day I will see again.

Preface

Papers written by the author comprise parts of chapter 3 (McHale and Husbands [2004c] and McHale and Husbands [2004b]) and chapter 4 (McHale and Husbands [2006]).

Software written in support of research described in this thesis comprises three iterations of evolutionary robotics simulation software plus a generic software application framework. This application framework is briefly described in appendix A. It was designed from the start to support multi-threading, taking advantage of the multi-core processors available in modern computer hardware. It is intended for use as a generic simulation platform supporting; real-time visualization and charting, neural network simulation, large-scale parameter editing, automatic serialization and simulation batch processing. It makes use of a novel meta-program language which provides us with a means of composing complex applications from discrete plug-ins through a simple script. Over 80 console and 60 windows based test applications have been written, totalling in excess of 350,000 lines of code. All this is the sole work of the author.

Additionally, source code written in the C programming language for the GasNet model was provided by Phil Husbands. This was rewritten by the author in C++ for use in parts of experiments described in chapters 3 and 4. The Autosim dynamics package was used in the simulation of bipedal locomotion described in chapter 3. This is partially based on earlier work conducted in collaboration with Chris Buckley. The comparative study of quadrupedal locomotion described in chapter 3 makes use of the ODE open source rigid body dynamics simulation library. Later robot experiments employ NVIDIA's PhysX

rigid body dynamics simulation package (chapters 4 and 5). Robot models used in chapters 3, 4 and 5 were constructed by the author using Autodesk's 3D Studio MAX. Exporters and parsers were written by the author to support this. The Agent and Neural Graph Library (employed in chapters 9 and 10) was written by the author, but makes use of the Lemon Graph Library internally.

Acknowledgements

Phil Husband's work on GasNets and his founding role in the field of Evolutionary Robotics inspired the work reported in this thesis. As my supervisor, his experience and expertise in this area provided me with an overall context within which the relative importance of ideas could be assessed and a general direction of research determined. Without his considerable patience, encouragement and flexibility this thesis would never have been completed. I am also indebted to Inman Harvey and all the other lecturers on the MSc in Evolutionary and Adaptive Systems at the University of Sussex, for their excellent introduction to this field of study. Andy Philippiddes and Seth Bullock provided me with valuable feedback in response to my viva voce resulting in considerable improvements to this thesis.

Adaptive Networks for Robotics and the emergence of Reward Anticipatory Circuits

Gary McHale

School of Engineering and Informatics, Department of Informatics
University of Sussex

A thesis submitted for the degree of
D.Phil in Computer Science and Artificial Intelligence

September 30th, 2011

Currently the central challenge facing evolutionary robotics is to determine how best to extend the range and complexity of behaviour supported by evolved neural systems. Implicit in the work described in this thesis is the idea that this might best be achieved through devising neural circuits (tractable to evolutionary exploration) that exhibit complementary functional characteristics. We concentrate on two problem domains; locomotion and sequence learning. For locomotion we compare the use of GasNets and other adaptive networks. For sequence learning we introduce a novel connectionist model inspired by the role of dopamine in the basal ganglia (commonly interpreted as a form of reinforcement learning). This connectionist approach relies upon a new neuron model inspired by notions of energy efficient signalling. Two reward adaptive circuit variants were investigated. These were applied respectively to two learning problems; where action sequences are required to take place in a strict order, and secondly, where action sequences are robust to intermediate arbitrary states. We conclude the thesis by proposing a formal model of functional integration, encompassing locomotion and sequence learning, extending ideas proposed by W. Ross Ashby.

A general model of the *adaptive replicator* is presented, incoporating subsystems that are tuned to continuous variation and discrete or conditional events. Comparisons are made with Ross W. Ashby's model of ultrastability and his ideas on adaptive behaviour. This model is intended to support our assertion that, GasNets (and similar networks) and reward adaptive circuits of the type presented here, are intrinsically complementary. In conclusion we present some ideas on how the co-evolution of GasNet and reward adaptive circuits might lead

us to significant improvements in the synthesis of agents capable of exhibiting complex adaptive behaviour.

Contents

PERMISSIONS

The following diagrams were reproduced with permission from their respective publishers;

- Figure 6.1 was adapted from Wilson [2004] and reproduced courtesy of Oxford University Press.
- The source for figure 6.3 was Nambu et al. [2002], reproduced courtesy of Elsevier Limited.
- The source for figure 8.2 was Schultz [1998], reproduced courtesy of The American Physiological Society.
- The source for figure 11.1 was Schultz et al. [1994], reproduced courtesy of The MIT Press.
- The source for figure 13.1 was Doya [2000], reproduced courtesy of Elsevier Limited.

List of Figures

Chapter 1

Introduction

1.1 Overview

Work described in this thesis has one underlying goal. That is to develop techniques and devise models that will allow us to create artificial brains capable of supporting complex behaviour in physically simulated robots through the evolutionary robotics methodology. This involves emulating both the *process* and *products* of biological evolution. We do this through abstractions of evolution in algorithms (the *process*) and abstractions of neural processes through the use of artificial neural networks (the *product*). Implicit in the approach taken by this thesis is the idea that we may be able to achieve significantly more complex behaviours than previously demonstrated, through the use of neural network circuits that exhibit *complementary* functionality. Specifically, we explore GasNets and other adaptive networks in support of locomotion and then seek to create complementary circuits that support reinforcement learning.

The novel contribution of this thesis is in the creation of a neuron model and corresponding circuits that exhibit reward adaptive behaviour. Whilst the formalisms of reinforcement learning underpin our understanding of how such systems may work in the brain, conventional solutions draw their approach from *dynamic programming* models and control theory. In contrast, we propose a biologically inspired model based on the role of dopamine in the basal ganglia. We seek to explain the workings of biological reward adaptive systems through

distinct neural modes that support action **selection**, **exploration** and **exclusion** (the active suppression of a specific selection option). The resultant model shares many similarities with that of the striatum and in particular the medium spiny neurons found therein.

The neuron model is computationally efficient and the resultant circuits are simple. The simplicity of these circuits is particularly appealing, since our goal was to devise models that are tractable to evolutionary exploration. The characteristic ability of these circuits to learn beneficial action sequences is complementary to that of GasNets and other adaptive networks that support legged locomotion explored in the earlier stages of this thesis. It is hoped that this thesis lays down some of the foundations necessary to support the co-evolution of these two distinct circuit types in future work.

Overall, this thesis strongly adopts the *biologically driven modelling* approach displayed in figure 1.1. By this we mean that our preference is to look at biological systems and create models which are intended as suitable abstractions of these systems. By *suitable*, we are referring in this context to models that are computationally efficient and yet capture some significant feature of the biological system that we are seeking to emulate. Generally in evolutionary robotics the models chosen are less biologically realistic than those employed in computational neuroscience, but more biologically plausible than neural network models generally employed in connectionist approaches to machine learning. Computational efficiency is of particular importance due to the onerous simulation requirements required by the evolutionary method.

1.2 Structure of the thesis

Figure 1.2 illustrates the structure of the thesis. The schemtic illustrates the principle concepts addressed in each chapter.

The key contributions of each chapter are summarized as follows;

- **Chapter 2:** This chapter introduces the basic concepts of evolutionary robotics. It describes in some detail the emergence of this field, reviews work carried out by the principal research centres and researchers, and

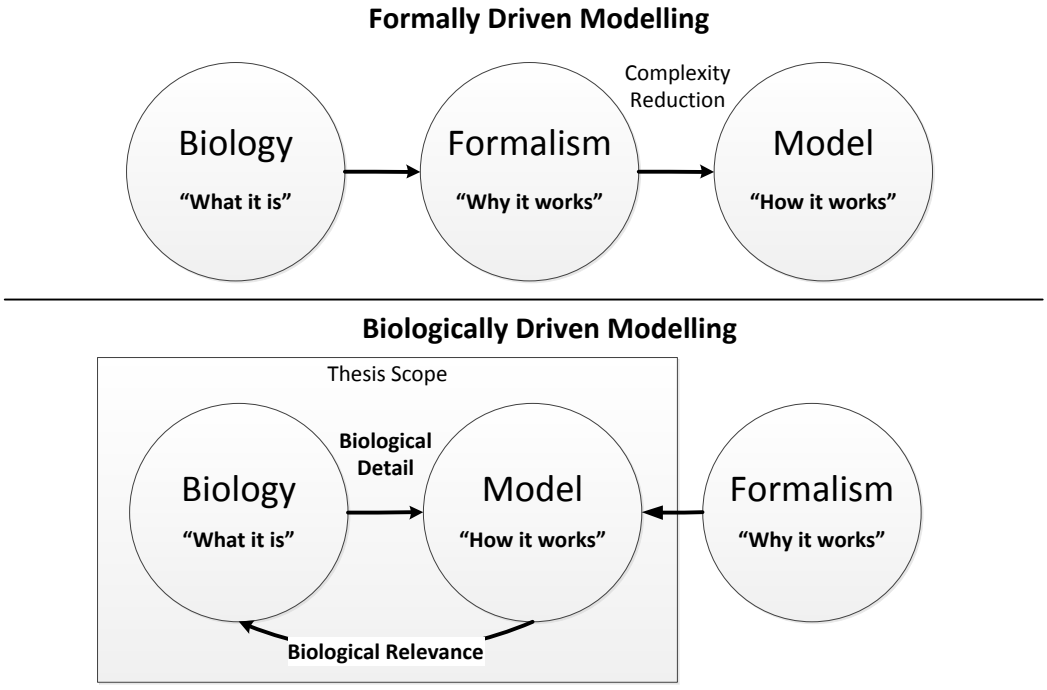


Figure 1.1: Philosophical Approach and Scope of the Thesis

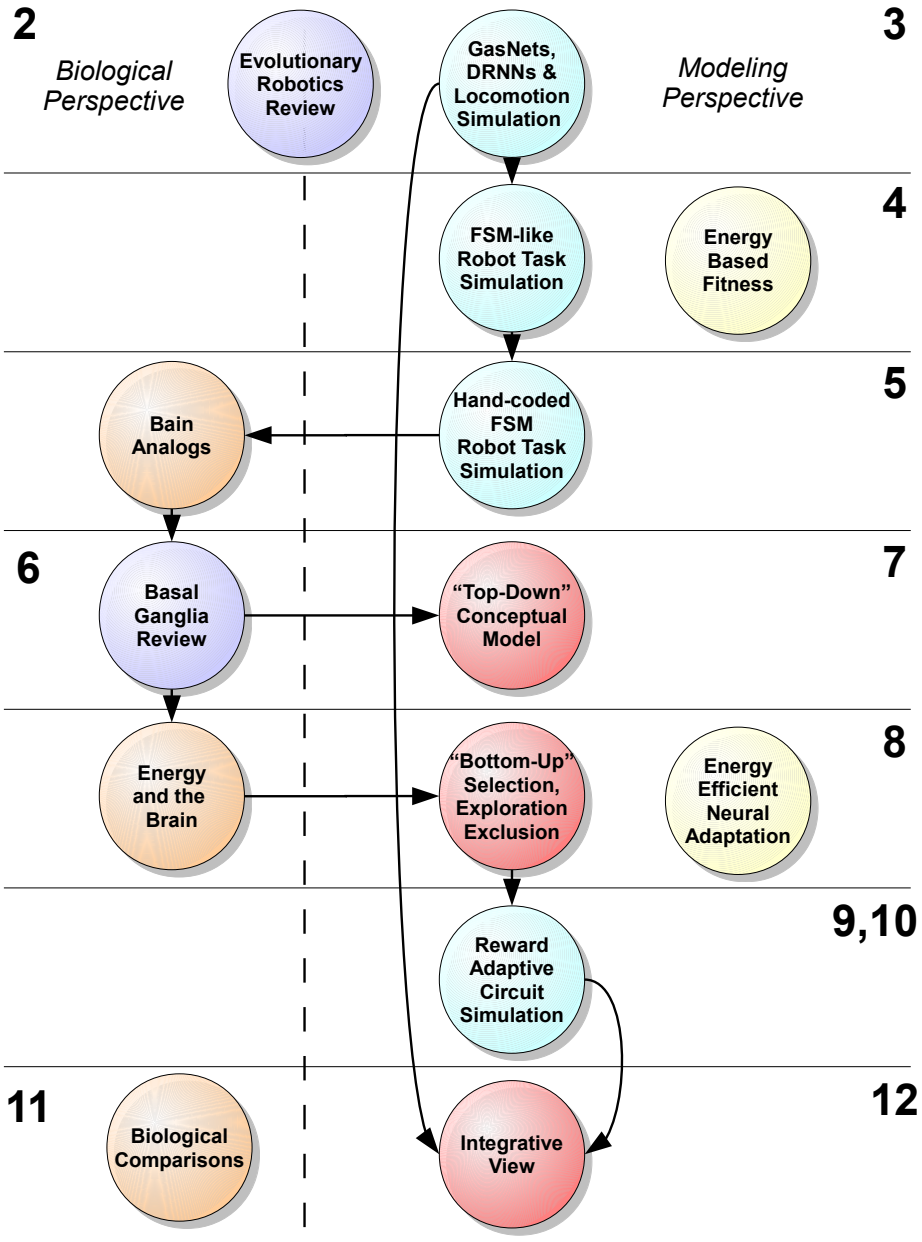


Figure 1.2: Thesis structure by chapter

concludes with the reasons for the approach outlined in this thesis.

- **Chapter 3:** This thesis commences with comparative study of the principal neural network models suitable for parametric encoding in a form that is tractable to evolutionary search. We demonstrate how GasNets and other adaptive networks are suitable for the generation of control circuits to support locomotion in legged robots (bipedal and quadrupedal).

- **Chapter 4:** This chapter describes the application of these same techniques to a ball-collection task. On order to shape the course of evolution, the energy expended by the robot was included in the fitness function. The idea was to promote implicitly *intelligent* behaviour through penalizing evolved solutions that relied upon brute-force (highly energetic) solutions to this problem. Despite this, GasNets were found to be less suitable for this task than they were in supporting locomotion. The ball-collecting task is typical of a class of problems that tractably through the decomposition of the task into smaller sub-tasks, where each sub-task requires a sub-set of sensor data and employs a sub-set of motor control outputs. In video games this kind of problem is typically addressed by the construction of an finite-state machine model. The lack of amenability to discovering such solutions via the evolution of GasNets led us to consider what alternative neural network formulations might be appropriate. This represents a significant divergence from the prior line of enquiry.

- **Chapter 5:** A decision was made to hand-code a solution to the ball-collection task. The goal here was to try and gain some insight into architectural features and data flow patterns that might be of particular relevance in solving this problem. We then compared such features of this hand-coded model to those present in biological systems. The basal ganglia was identified as potentially significant in solving such tasks.

- **Chapter 6:** A review of the basal ganglia was conducted, with a principal focus on the role of dopamine in reward adaptation.

- **Chapter 7:** Following this review, we constructed a *top-down* model of the basal ganglia and its associated systems. This was largely done as thought

exercise to understand the important features that such a system might incorporate. Whilst the ensuing model is appealing in itself, it represents a *goal* or *target* of evolutionary search. Our requirement is to discern how such a system can be evolved from less primitive circuits. One of the primary tenets of evolutionary robotics is that the user should not pre-determine the architecture of the system, but support the exploration of architectural space through lower-level neural models. However, work in this chapter is justified in that it helps us to appreciate the possible solution space of architectures that should be accessible via our evolutionary methodology

- **Chapter 8:** With this in mind our focus shifts to models of neurons and neural circuits that might be suitable substrates for evolutionary search. In this chapter a new neuron model is proposed and a primitive circuit based on this model is introduced. Whilst the previous chapter takes a *top-down* view this chapter seeks to take inspiration from the purported role of dopamine signalling from a *bottom-up* perspective. We commence by first seeking to identify fundamental principles that might allow us to explain certain phenomena observed in dopamine signalling. A hypothesis is constructed based on the idea that energy efficiency plays a significant role in determining neuron and signalling characteristics. This assumes that evolution preferentially selects energy efficient systems. Neural inhibition as it exists in our model, occurs when a beneficial signalling pathway (i.e. one which takes part in an identified chain of input-output pairings that lead to a reward) has undergone adaptation. The effects of this inhibition are to minimize the energy cost of signalling and to remove such pathways from competitive activation when their inputs are inactive. One of the appealing aspects of this model is that it may provide us with some explanation regarding the prevalence of inhibitory pathways in the basal ganglia.

- **Chapter 9:** We then carry out a number of experiments that correspond to incremental adaptations in simple reward circuits based on this novel neuron model. The problem class encompasses two types of sequence learning; *hard* sequences, where actions taken in a unique sequential order elicit a reward, and *soft* sequences where a reward is elicited based on an action sequence

that might include a number of neutral intermediate actions

- **Chapter 10:** The principal results are described, with two circuit variants being identified as particularly suitable for solving both *hard* and *soft* sequence problem respectively.

- **Chapter 11:** Biological plausibility is a cornerstone of the evolutionary robotics methodology. Whilst it is recognized that we need to make some sacrifices in the area of biological realism for the sake of computational expediency, an attempt is made to seek to discover abstractions that in some way capture some significant features of the biological systems that we are seeking to emulate. In this chapter we compare our model with biological systems. Our model shows significant similarities with the characteristics of striatal medium spiny neurons (MSN) of the basal ganglia.

- **Chapter 12:** Since the work described in this thesis took place over an extended period, alternative approaches to modelling reinforcement learning have emerged. We first recap some of the advances that have been made in this area, in addition to reviewing some of the results of other researchers that represent a continuation or extension of work conducted in the early part of this thesis. The second part of this chapter seeks to propose an integrative view on how GasNet (and similar models) might work in conjunction with the reward adaptive circuits.

The thesis is best conceived as comprising four parts; firstly, an exploration of existing evolvable neural network models; secondly, the determination of shortcomings in the use of these models; thirdly, the creation of biologically inspired neural network models which undergo adaptation to rewards; and fourthly, an integrative perspective on the relevance of these models to evolutionary robotics their biological plausibility.

Chapter 2

Evolutionary Robotics

2.1 Introduction

Evolutionary Robotics comprises three key elements. The first is goal oriented. It seeks to discover methodologies for developing control systems for real or simulated robots for use in practical applications. The second is methodological. It draws inspiration from the biological systems that underpin intelligent behaviour and crucially employs evolutionary approaches to the generation and development of models which seek to capture the essence of these biological systems. The third is theoretical. It is hoped that through the pursuit of this goal, using methodologies that are based on natural systems, that we are able to gain a deeper understanding of the principles and processes that have resulted in the emergence of intelligent life.

We start this chapter by outlining some of the basic ideas that underpin the evolutionary robotics methodology, focussing on the processes of genetic adaptation. The synthesis of ideas from; cybernetics (section 2.3.3), evolutionary theory and the algorithms that seek to model it (section 2.3) and neural networks (section 2.3.2), has precipitated what is arguably one of the most conceptually rich areas of modern science. In order to explain why this is so, we spend some time delineating the historical emergence of this field (section 2.3). Following this we carry out a selective review of the work carried out at the major research centres in Europe (Sussex, CNR and EPFL) and of the research conducted by prominent

individuals in the USA (section 2.4).

This chapter constitutes a review of evolutionary robotics and the inspiration for the work described in this thesis. We also carry out a review of literature that relates to the basal ganglia (see chapter 6) since this underpins much of the described in the second half of this thesis. The research described in this thesis was undertaken over a prolonged period of time (largely from 2004 to 2012). Whilst the work undertaken in 2004 is based on ideas prevalent in evolutionary robotics at that time, later work described in this thesis diverges somewhat from the path taken by other researchers. We shall summarise these developments in a later chapter (see chapter 12).

We conclude this chapter by explaining the focus of this thesis; that of determining how we might support behavioural complexity greater than that which is achievable through current methods.

2.2 Basic Concepts

Evolutionary Robotics employs evolutionary algorithms to obtain its results. There are a wide range of evolutionary algorithms, referred to alternatively as *evolutionary programming*, *genetic algorithms*, *evolution strategies* and *genetic programming*. We shall review the differences between them in section 2.3.1.3. What we wish to do here, is explain the common ideas that these algorithms embody using a set theoretic approach. We shall define some terms differently from how they are commonly used, but seek to highlight these differences where appropriate.

2.2.1 Genotype Space

The notion of a genotype is central to evolutionary algorithms. In biology a genotype represents the totality of genetic information associated with an organism. It comprises all the chromosomes that the individual possesses. These structures contain alleles (alternative forms of genes, or allelomorphs) present at specific loci along their length. It is therefore not just the chromosomes themselves, but the particular combination of alleles comprising them that define the genotype of an

individual.

It is sometimes said that a genotype *codes for* the individual. However, this represents a computational analogue that is only appropriate for the simplest of explanations. The sense in which a computer program is coded such that it defines its operation in a deterministic fashion is *not* analogous to the way in which a genotype determines an individual. The simple reason being that the genotype underpins a number of adaptive processes, including development and learning. It is the entirety of these processes that determine the phenotype; the form, attributes and behaviour of the individual organism. This occurs as a result of a coupling of these adaptive processes with the environment and the entities that inhabit it. By this we mean that the environment itself is an input to the functions that determine an individuals characteristics. The contributory nature of these adaptive processes, that lend a degree of *plasticity* to the phenotype, has no analogue in a computer program.

It is based on the differential variation of phenotypes within a population that selection acts, thus determining the frequency of alleles in the population in subsequent generations. The notion of a genotype space is central to the exposition of evolutionary algorithms. A genotype maps to an individual (the phenotype of a genotype), as if it contains the parameters of a function. It is through the exploration of this parameter space that we are able to discover individuals with particular traits that are deemed beneficial. In the case of evolutionary robotics these are traits that are particularly suited to the tasks that we wish our robots to perform. We are able to explore this space through the action of recombination and mutation operators on the parameters encoded by the genotype.

A verbal exposition using terms heavily laden with biological significance lacks the degree of clarity that we seek in attempting to elucidate the essential features of an evolutionary algorithm. In biology ambiguities can arise since the term *gene* is on occasion used to refer to an *allele* (without differentiating between type or value) and on other occasions refers to a specific locus (or group of loci) on a chromosome. For this reason we use a set theoretic approach. We seek to abstract these biological concepts in a fashion that is relevant to explaining the underlying structure of evolutionary algorithms. This requires a definition of terms; *gene*, *allele type*, *allele value*, *allele group*, *chromosome*, *genotype*, and *set*

of genotypes, in a set theoretic sense, broadly comparable to their use in biology. The exposition described here is a primitive attempt to employ types in naive set theory. A more rigorous approach might avoid this completely and use type theory formalisms. Whilst it may not be particularly rigorous, it should at least help to clarify the conceptual notions associated with evolutionary algorithms as they are commonly used in evolutionary robotics. The essential features of this approach are illustrated in figure 2.1.

The principal dimensions of this explanation are those of *type*, *value* and *structure*. As an analogy we imagine a row of wooden blocks. Each block has a hole in it. The shape of the hole varies from block to block (e.g. square, rectangular, circular). Multiple blocks may have the same shaped hole. We also have a number of bricks. We can partition these bricks into groups based on their shape. The shape of the bricks is such that they may be inserted into the appropriately shaped holes present in the wooden blocks. The bricks are painted different colours. We are able to insert any coloured brick into a wooden block, providing that it has the appropriately shaped hole.

In this analogy, we can refer to the shape of a brick and the same shape of hole in a wooden block as a *type*. We can refer to the colour of each brick as a *value*. We can refer to a particular arrangement of wooden blocks as a *structure*. The *structure* corresponds to a *syntax*. The colour of bricks inserted into these wooden blocks, endows the structure with *values*. These *values* correspond to *semantic* information, in a system where the particular arrangement of structure and *values* determines the characteristics of a subsequent process.

This concrete example helps us to appreciate the sense in which we are to use the terms; *type*, *value* and *structure* in the following definition of terms;

- **Gene (G):** We define a *gene* as comprising two elements; an *allele type* element and a set that holds a single *allele value* element. The value element is an empty set. This makes the *gene* analogous to an empty wooden block. The *allele type* element is analogous to the shape of the hole. The second member, the set which holds the value element, is an empty set. This is analogous to the empty hole in a wooden block. In our definition a *gene* represents a structural component with a specific type property, but remains absent of semantic value.

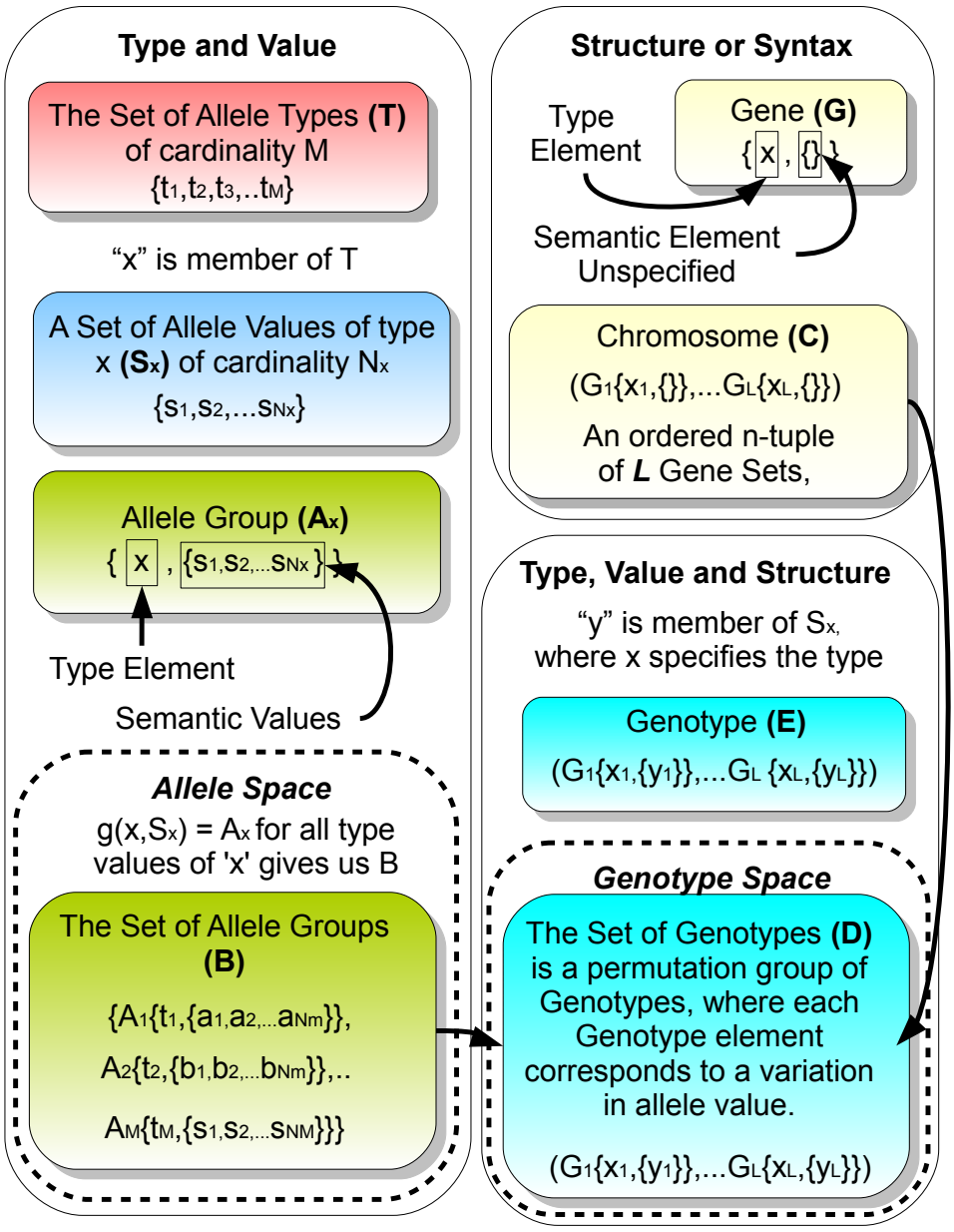


Figure 2.1: Type, Value, Structure and Genotype Space

- **Chromosome (C):** A *chromosome* is defined here as an ordered tuple of dimension L, where each element is a *gene*. Whilst in biology an organism may possess multiple chromosomes we use the term *chromosome* in a way that assumes that a single organism only possesses one of them. In an extension of the ideas implicit in our definition of a *gene*, a *chromosome* is also a structural entity devoid of semantic information.

- **Set of Allele Types (T):** There exists a set comprising elements, each of which corresponds to a separate *allele type*. The *type* of an allele is analogous to that of the shape of a brick, and the shape of the hole in which the wooden block in which it can be placed. In figure 2.1 we use the dummy variable x to signify a member of the set T.

- **Set of Allele Values (S):** The *value* of an allele is analogous to that of the colour of the brick. For each *allele type*, there exists a corresponding set S, containing all the possible *values* that an allele may take. In figure 2.1 we use the dummy variable y to signify a member of the set S_x, where x is a dummy variable corresponding to an *allele type*, taken from the set T.

- **Allele Group A:** An *allele group* is a set that has as its first member an *allele type* x, and its second member a set of all possible *values* that have this type as a property (i.e. the set S_x).

- **The Set of Allele Groups (B):** The set of *allele groups* is the comprising all possible *allele groups*.

- **Genotype (E):** A *genotype* corresponds to the result of an binary operator that takes as its arguments the *set of all allele groups* and a *chromosome*. This operator iterates through the each *gene* element of the *chromosome*. It identifies the *allele type* member of the current *gene*, and from the corresponding *allele group* (i.e. that which has the same *allele type* as its type member), extracts an *allele value* (from the set of *allele values* embedded in the *allele group*). A *genotype* is analogous to a row of wooden blocks, where the hole in the each block has been filled with a brick of the appropriate shape. It therefore comprises both syntactic and semantic information.

- **The Set of Genotypes (D):** We can imagine a set operator that is analogous to cartesian product, where we multiply a *chromosome* (C) with the *set of allele groups* (B). In a given *chromosome*, the elements that it contains (its *genes*), each have a specified *allele type*. Such a cartesian product would ensure that each possible allele value, for each type, was present in a *set of genotypes* (D). This set D corresponds to all the possible permutations of *genotypes* (E), where allele values are varied, but the *gene* elements of the *chromosome* remain constant. This set encompasses the entirety of all the possible arrangements of *allele values* in the specified *chromosome*. To use the blocks analogue, the *set of genotypes* (D) corresponds to a set of elements, where each element corresponds to a chain of wooden blocks with all holes filled by colored bricks. There exists all the possible permutations of coloured bricks inserted into these blocks (of defined type and order), such that there each element is unique, and that all possible permutations are represented.

What we are keen to do here, is to differentiate between concepts which primarily relate to structure or syntax (the *gene* and *chromosome* in our terminology), from those which relate to semantic value (in our case *allele value*). We are then able to create derivative structures, through operations that rely upon a shared property of *allele type*; such as the *allele group* and the *genotype*. Each *gene* has an associated *allele type*. Each *allele value* has an associated *allele type*. We end by defining a structure (D) that encompasses all the possible permutations of *allele values* consistent with the structure of a *chromosome*. Now that we have defined a what we mean by a genotype space (i.e. the parameter space corresponding to the *set of genotypes*), we are in a position to outline the structure of an evolutionary algorithm.

2.2.2 The evolutionary algorithm loop

A schematic for the evolutionary algorithm loop is shown in figure 2.2.

The evolutionary algorithm starts by sampling elements from the *set of genotypes* (D). This takes place via an *initialization operator*, which generates members of the *population* (H). Members of the *population* (H) comprise two elements;

the first of which is an element taken from the *set of individuals*, and the second a set that corresponds to a single *genotype* (E). Although the element taken from the *set of individuals* (I) might simply correspond to an index (into an array of data structures for example), there are alternative implementations where each element might correspond to a more complex structure (such as a label denoting species as well in addition to a unique index for example).

The elements of the *population* (H) represent the substrate upon which the evolutionary algorithm acts. Typically it is only the distribution of *allele values* amongst the *population* members that change over time. In more complex algorithms, the structure of the *chromosome* is also subject to modification, as *genes* are added or removed. Whilst the number of *population* members will commonly remain constant in over the course of the algorithm iterations, this too may also be subject to change in alternate implementations. The *population*(H) is a superset containing a *tournament set* (R), which in turn is s superset containing the *breeding set* (V). Members of the *tournament set* (R) are mapped to a *set of trial phenotypes* (Q). It is in this mapping that the semantic significance of the *allele values* contained in the *genotype* plays a role. The following explanation highlights the structure of an evolutionary algorithm by focussing on set operators (used to generate sets) and functions (that map elements of one set to another).

The characteristics of an evolutionary algorithm are determined by the implementation of the operators and functions employed, the principal of these are;

- **(D, I to H) Initialization Operator:** The *initialization operator* generates the elements of the *population* (H). It does this through sampling from the *set of genotypes* (D), combining each *genotype* with an element from the *set of individuals* (I).

- **(H to R) Tournament Operator:** The *tournament operator* is essentially a unitary operator that determines which of the members of the *population* are to be included in the *tournament set* (R).

- **(R to Q) Ontogenetic Function:** The *ontogenetic function* is that which maps elements of the *tournament set* (R) to the *set of trial phenotypes* (Q). In the case of evolutionary robotics, the *genotype* contained in the

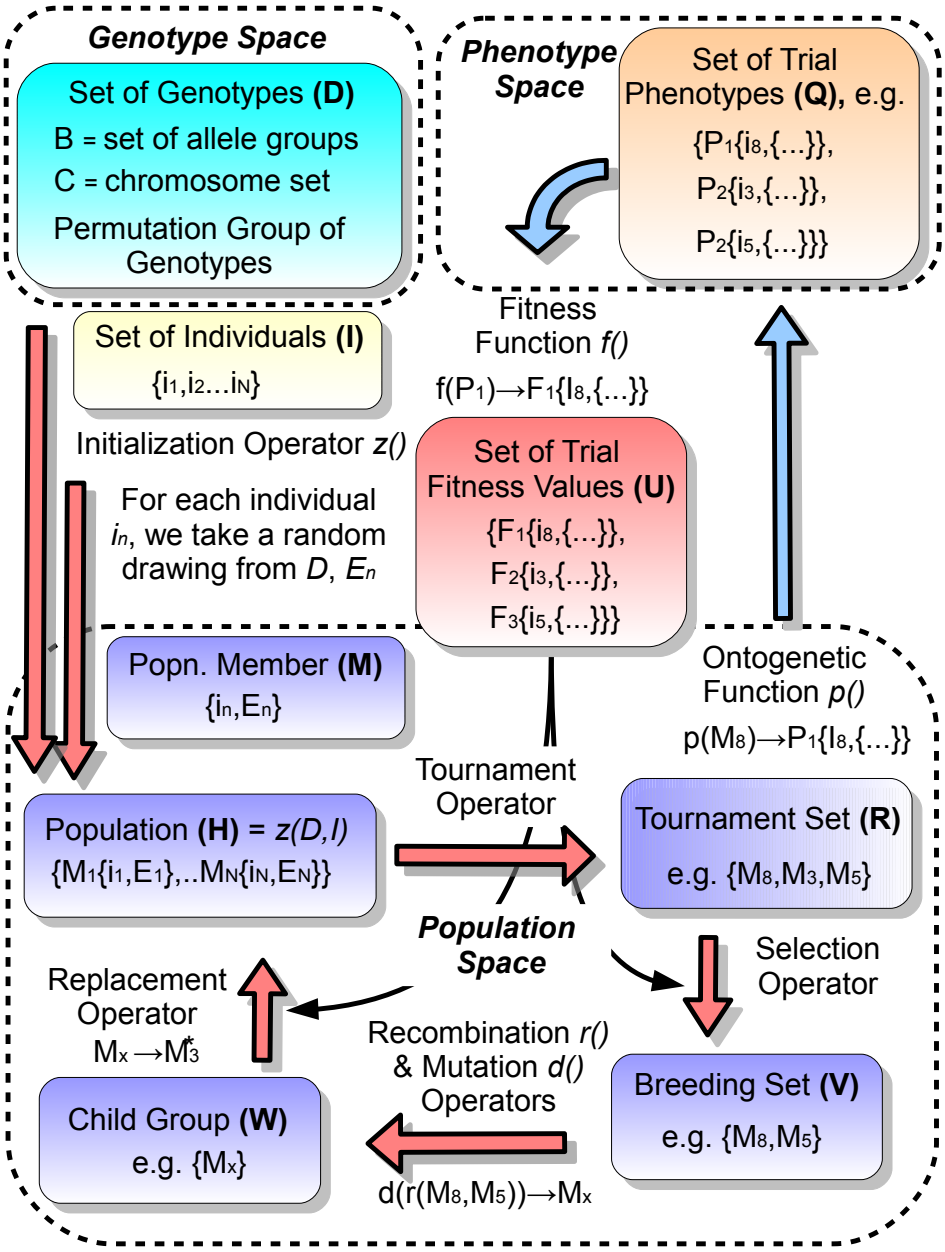


Figure 2.2: The evolutionary algorithm loop

population member usually maps to an artificial neural network (although in more complex implementations robot morphology might also be determined). This mapping takes place via an *ontogentic function* that treats *allele values* as parameters.

Sometimes we refer to *direct* or *indirect encoding*. Direct encoding is to be employed where the *allele values* correspond to explicit features or parameters of the neural network model without modification. *Indirect encoding* is employed in methodologies where the *allele values* comprise parameters of a generative model. The generative model might be some abstraction of a developmental process. Algorithm design has as its goal the efficient exploration of parameter space, for the purposes of discovering the fittest individuals. As a consequence, the decision to employ *direct* or *indirect* coding cannot be determined in isolation. Other significant factors include the recombination and mutation operators, as well as the functional characteristics of the instantiated neural network phenotypes.

- **(Q to U) Fitness Function:** The *fitness function* is a mapping from each member of the *set of trial phenotypes* (Q) to a member of the *set of trial fitness values* (U). In evolutionary robotics, the neural network is instantiated in a robot (corresponding to a member of the *set of trial phenotypes*) and a metric of the success in completing a task is used to determine a value that is a measure of the extent to which the robot successfully completed the task. It is this value, (associated with the population member index) that is stored in the *set of trial fitness values* (U). These values are important, since the are used to parameterize subsequent set operations; the *selection* and *replacement operators*.

- **(R to V) Selection Operator:** The *selection operator* determines which members of the *tournament set* are to be included in the *breeding set* (V). The *set of trial fitness values* (U) is employed in determining the outcome of the *selection operator*. In its simplest form, we simply rank *tournament set* members according to their corresponding *fitness values* (F). The fittest are then allowed to reproduce.

- **(V to W) Recombination and Mutation Operators:** These operators are key in determining how new genotypes are created. The *recombination operator* (r) is a binary operator that requires two elements from the *breeding set* (V) to generate a new member of the child group (W). The *recombination operator* (r) does this through a principled mixing of the semantic information contained in the *genotypes* (E) of both *parent breeding set* members.

  Such an operator might for example employ *single-point crossover*. This is where a locus on the *chromosome* (C) is chosen at random. This locus is used to partition the *chromosome* into two intervals; from the start of the chromosome to the locus, and from the locus to the end of the chromosome. The *allele values* corresponding to that of the first interval from one parent, and the second interval from the other parent are copied over to the progeny (member(s) of the *child group* (W)).

  The *allele values* of this *child group* (W) member are then subject to further modification by a *mutation operator*. This is a unary operator that changes some or all of the *allele values* present in the child group (W) member's genotype (E) in some principled fashion, usually involving random perturbation.

  The recombination operator is intended to be analogous to processes employed in sexual reproduction. However, not all genetic algorithms employ a recombination operator. Clonal genetic algorithms simply mutate the genotypes of the breeding group, in a fashion that is analogous to asexual reproduction.

- **(W to H) Replacement Operators:** Finally the *replacement operator* ensures that the least fit member(s) of the *tournament set* are replaced by members of the *child group* (progeny of the *breeding set* (V)). The set of trial fitness values (U) is used to determine which population member(s) present in the *tournament set* (R) correspond to members of the *set of trial phenotypes* (Q) that performed the least successfully. The index of the population member(s) so identified, is used to determine which member of the *population* (H) has its allele values replaced by those of the child group

(W) member .

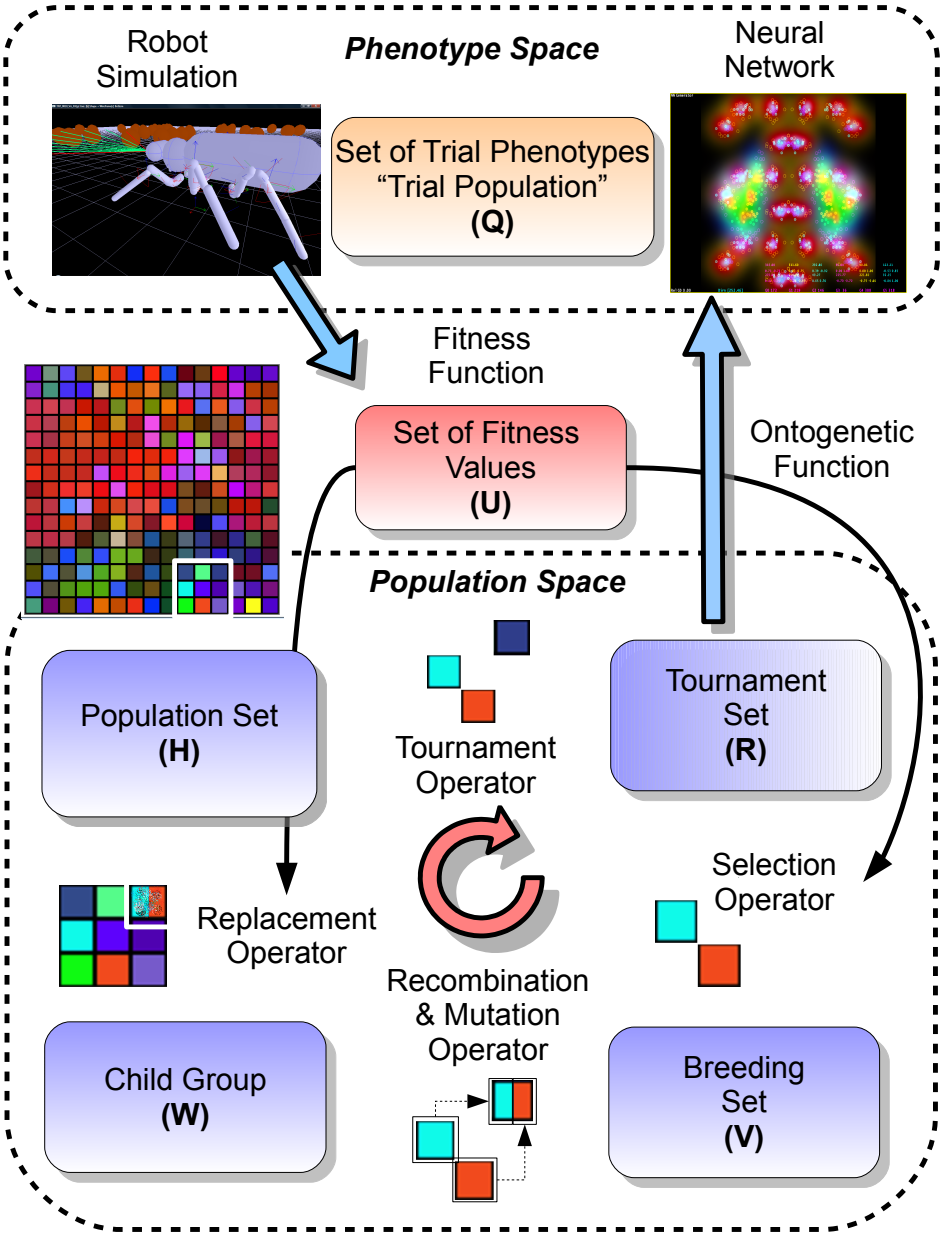


Figure 2.3: Evolving Robot Brains

2.2.3 Client-Server Architectures

Figure 2.3 illustrates how these concepts are applied in evolutionary robotics. We shall address these issues in more detail in section 2.3.1.4. Whilst many problem domains make use of evolutionary algorithms that match this pattern there are two specific areas of implementation that are highly specialized when applied to evolutionary robotics. These are; firstly the *ontogenetic function* which maps the genotype to the neural network phenotype and secondly the *fitness function*.

The fitness function is based on a some measure of performance of the robot in the simulation and is therefore implicit. Unless the researcher is addressing collaborative or collective behaviour on the part of simulated robots, we can create a very simple computer architecture based on a *client-server* model for use in carry out the robot simulations used in the calculation of a fitness value using this measure. A typical *client-server* architecture is illustrated in figure 2.3. The role of the *server* is to execute the initialization, tournament, selection, recombination and mutation operators and to store the *population*. Once we have a valid *tournament set* the server can send *population member* data (including the *genotype* associated with this member) to a remote *client* computers. These client computers carry out the onotogentic mapping from the *genotype*(E) to the *phenotype set* (P). Typically this involves the instantiation of the neural networks for robot control. The client will then, carry out the trial simulation that allows us to calculate the genotypes corresponding fitness value. Once this has been done, the results can be returned to the *server*, where they are collated in a *set of trial fitness values* (U). The server then executes the selection operation on the *tournament set* (R), to determine which of the *tournament set* members are to be selected for breeding. Those selected comprise the *breeding set* (V). The server then executes the recombination, mutation and replacement operations. The distributed algorithm is shown in more detail in figure 2.5.

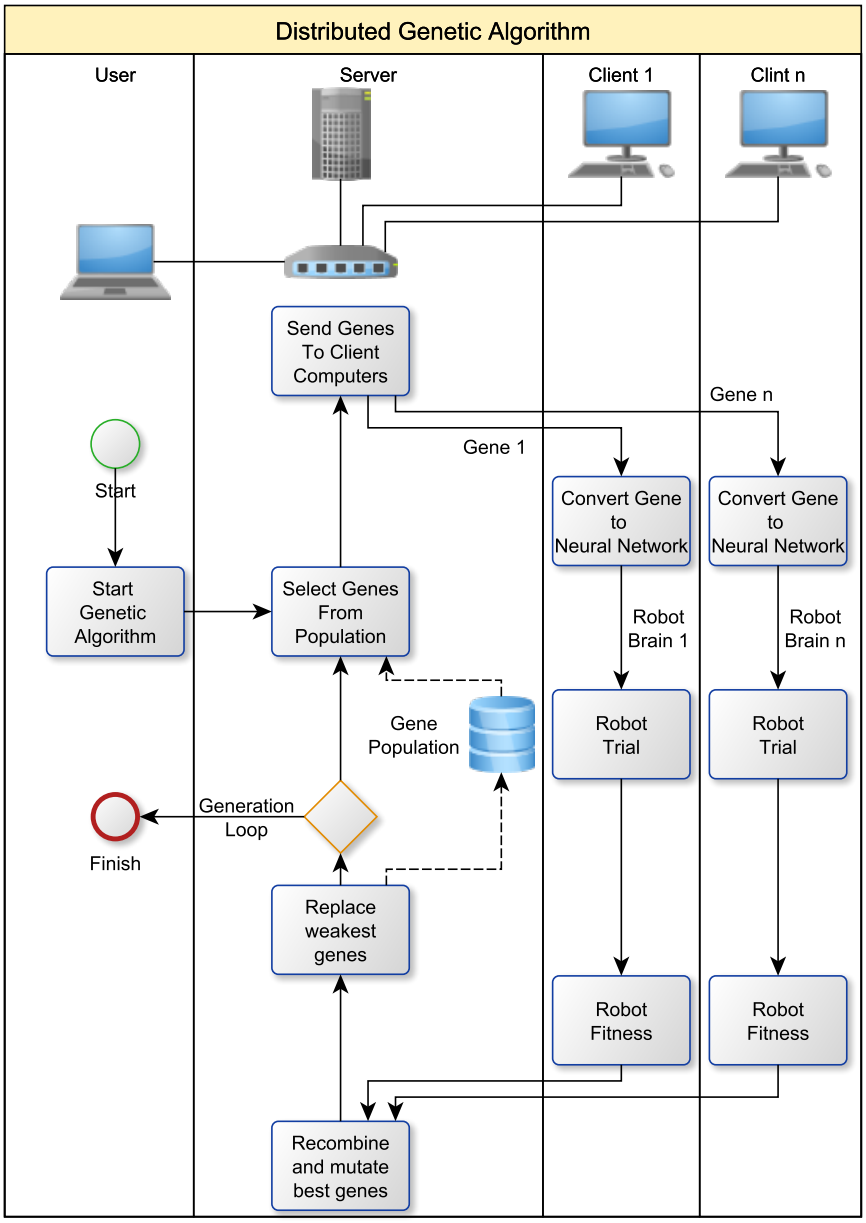


Figure 2.4: Client-Server Networks for Calculation

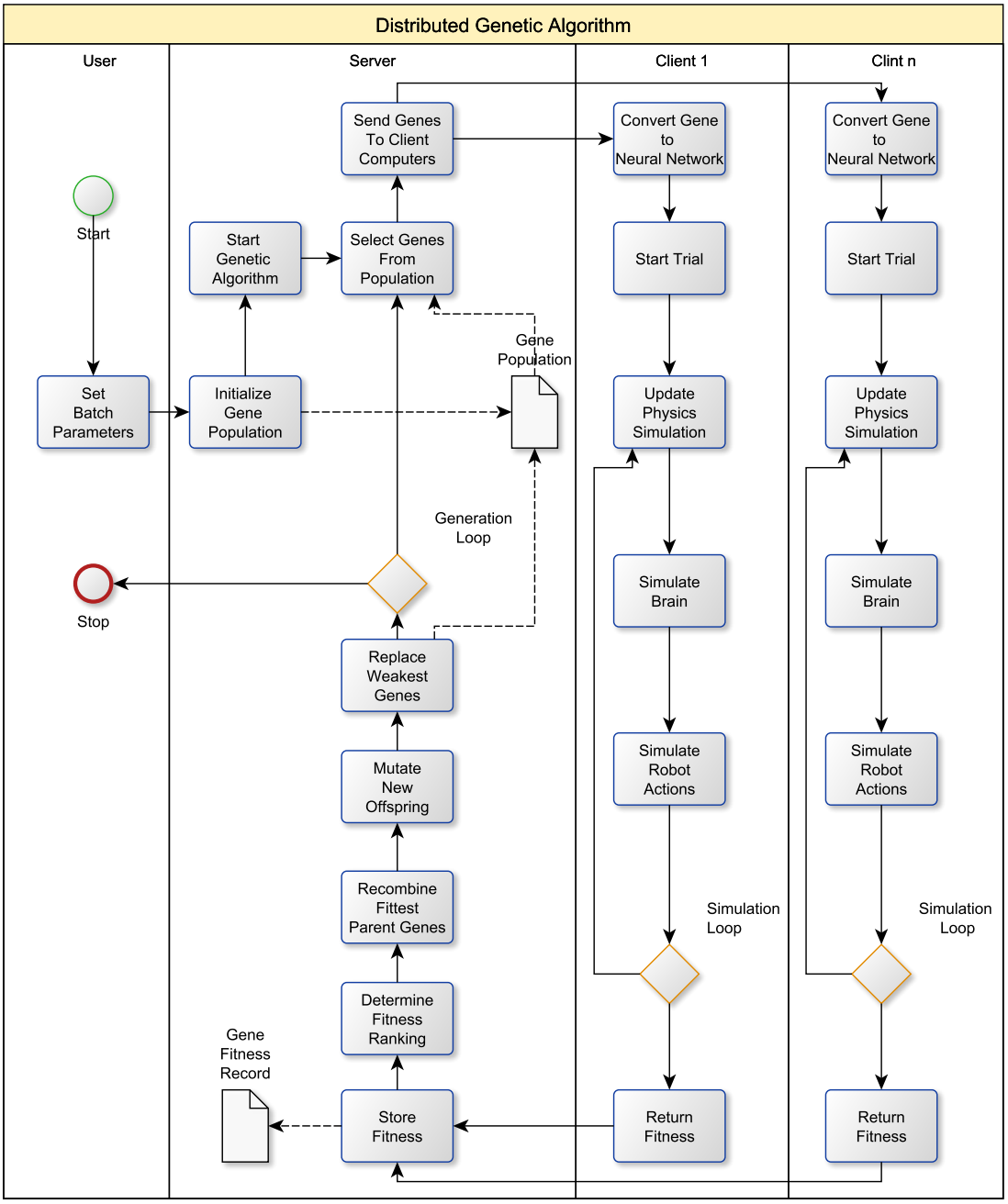


Figure 2.5: Distributed Evolutionary Algorithm

2.3 The Anastomosus of Evolutionary Robotics

An *anastomosis* is a structure comprising a network of streams which branch and reconnect (a term used in biology and geology). Warren Sturgis McCulloch used the term *anastomosus* when seeking to characterize the complex origins of cybernetics (Mcculloch [1974]). The image that he presumably had in mind was that of science and particularly cybernetics, as a flow of ideas and concepts that over time separate and recombine. What is true of cybernetics is even more true of the field of evolutionary robotics, since it encompasses cybernetics and its derivative streams in both thesis and antithesis (see figure 2.6 for a concept map of the major themes and ideas that have contributed to this field). One might justifiably argue that evolutionary robotics is one of the most conceptually eclectic of all the sciences, incorporating ideas from;

1. Evolutionary theory and computational algorithms inspired by natural systems.
2. Cybernetics and the theory of control systems.
3. Neural networks (from the differing perspectives of connectionism, neuroscience, neurobiology, neuropsychology).
4. The philosophy of artificial intelligence (largely building on ideas proposed by Heidegger).

Whilst evolutionary robotics is constructed from these foundations, it is also partially defined as the antithesis of the computational approach to artificial intelligence. The notion of *embedded cognition* (see section 2.3.4) is key in highlighting the contrasting approaches of these two fields and does much to frame the evolutionary robotics approach.

2.3.1 Evolutionary Theory

2.3.1.1 The Modern Synthesis

What is often referred to as the *modern synthesis* relies upon ideas described by Charles Darwin in his book *On the Origin of Species* (Darwin [1859]). His work

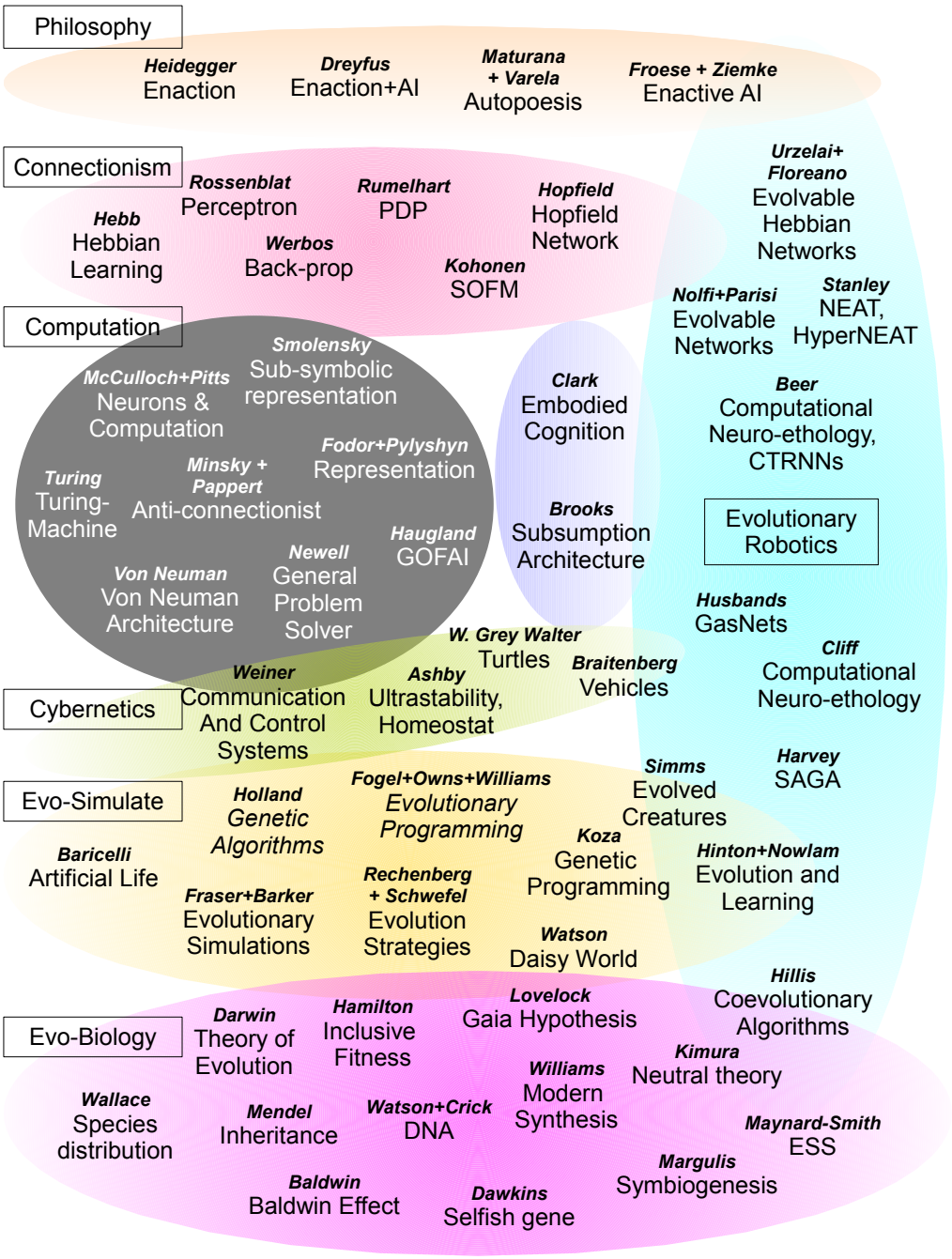


Figure 2.6: Evolutionary Robotics Concept Map (Computation as the Antithesis)

features some of the ideas previously expounded by Alfred Russel Wallace (Wallace [1855]). Darwin himself acknowledges that he was aware of Wallace's work (for a detailed treatment of this subject see Beddall [1972]). Wallace describes the gradual change of *form and structure* by species and the tree-like structure that these changes exhibit over time. However, there is one key element that is missing, namely the process by which this change occurs. Darwin's key contribution is the notion that organisms undergo a process of *natural selection* analogous to that of the process of *artificial selection* undergone by domesticated species due to selective breeding. It is the idea that through competition, changes take place in the distribution of characteristics in offspring as if *nature* were some arbiter of those that should survive and reproduce.

Darwin at the time was unaware of the existence of genes, nor the principles of inheritance that genes support. Gregor Mendel was the first to recognize that the inheritance of characteristics by offspring, does not take place through the blending of parental characteristics. Rather, certain characteristics are discrete and are inherited in their entirety from one parent or another (Mendel [1865]). The biological mechanism underlying this process was not identified until much later. James D. Watson and Francis Crick, identified the structure of DNA to be that of a double helix (for a historical treatment see Watson [1968]). The process of inheritance was thereby understood to rely upon genetic recombination and mutation (due to replication errors). Their work was based on X-ray crystallography work carried out by Rosalind Franklin (for an account see Maddox [2003]).

Modern evolutionary theory extends significantly on the ideas promoted by Darwin. It encompasses the idea that nominally altruistic activities are consistent with the persistence of a trait to the extent that it benefits those who share the same genes (Hamilton [1964]). In the narrowest sense where this relates to close family members this is termed *kin selection*, in the more general sense where individuals share the same ancestor this leads to the notion of *inclusive fitness*. The careful delineation of that which constitutes behaviour that will be selected for is an important theme in evolutionary biology with both Charles Williams (Williams [1966]) and Richard Dawkins (Dawkins [1976]) doing much to clarify these issues. Both of these authors argue against unsophisticated models

of *group selection* (Wynne-Edwards [1963]); the idea that altruistic behaviour will be *selected for* when groups engage in competition. Their criticism relies on the idea that altruistic traits (in excess of those implied by kin selection for example) incur a cost, and that less altruistic members of the group, who do not incur this cost, will proliferate. Simply put, altruistic populations are vulnerable to parasitic invasion since natural selection operates at the level of the individual rather than the group. Dawkins later introduced the notion that selection takes place, not simply due to the physical attributes of the individual, but also due to the effect that it has on its environment, referring to this as an *extended phenotype* (Dawkins [1982]).

It is acknowledged that social mechanisms potentially support other forms of altruism (see Alexander [1974] for an early review) through, for example *reciprocal altruism* (Trivers [1971]). Under these circumstances a social contract between two (genetically) unrelated individuals is supportive of altruistic acts to the extent that the individual can expect such acts to be reciprocated. However, such forms of altruism can be shown to rely upon the penalization of defectors (those who might unilaterally renege upon their social obligation). This in turn requires the existence of social practices that discriminate against those who do not penalize the defectors (assuming that is, that the enforcement of punishment is not cost free), in a potentially recursive fashion (see West et al. [2007] for a review of such issues). The extent to which social activity within a group is consistent with traits that will be *selected for* is an area which has received some elucidation through the application of game theory to social situations. John Maynard Smith deals with this subject comprehensively in his work *Evolution and Theory of Games* (Maynard Smith [1982]). It includes the notion of an *evolutionarily stable strategy* (ESS); a strategy that when adopted by individual population members, is impervious to displacement through invaders who might employ a variant of this strategy. This work is based partially on ideas elucidated by George R. Price (Smith and Price [1973]). The entirety of the ideas outlined here (in addition to some others) is often referred to in brief as *the modern synthesis*.

2.3.1.2 Further Developments

The ideas of natural selection, kin selection, inclusive fitness, evolutionary games theory and the clarification of the issues that relate to group selection (as elucidated by Williams and Dawkins), arguably constitute the core principles that comprise the modern synthesis. However, there are four key areas where the modern synthesis (sometimes referred to as neodarwinism) is subject to continuing analysis and debate, since;

1. It (arguably) understates the role of development and environmental feedback on the determination of traits.

2. It does not address the evolution of symbiotic mutualism and thus endosymbiosis

3. It understates the complexity of lineages by failing to take into account lateral gene transfer.

4. It is equivocal on the topic of multi-level selection.

Developmental biologists are particularly vocal on this first point (see for example Kitcher et al. [2001]). The orthodox response would be to take the position that phenotypic plasticity is a genetic trait and therefore is subsumed under neodarwinism.

The remainder of these points, relate to what is sometimes termed interspecific evolution (Hoffmeister and Martin [2003]), and are less easily dismissed. Symbiotic mutualism, is an area that has sometimes been avoided by neodarwinists due to the difficulties of explaining it within the standard model. Dawkins for example, focusses primarily on parasitic symbiosis (more readily explained under the Darwinist model) when he discusses symbiosis in *The Extended Phenotype* (Dawkins [1982]). Whilst it may be argued that mutualistic symbiosis is relatively rare in living organisms, its impact is not insignificant. The concept of endosymbiosis was proposed by Lynn Margulus (Sagan [1967], Margulis [1976]) to explain the emergence of eukaryotes through the symbiosis of prokaryotic ancestors. It is only relatively recently however that this notion has been accepted as part of the modern orthodoxy (Margulis [1995]).

The focus of the modern synthesis is on the individual, and the consequences of the actions taken by the individual that effect reproductive opportunity the subsequent survival of offspring. It is on the basis of this behaviour that selection is made. The collective behaviour of a group, can only therefore be interpreted as the macro effect of this constituent behaviours.

One might argue however, that whilst this might be true, it does not provide us with a parsimonious explanation of the behaviour of populations that co-adapt.

An example of such a system occurring in nature, is that of bacterial populations and the plasmid replicators that they host. Plasmids are parasitic replicators capable of carrying DNA between bacteria. Lateral gene transfer (Freeman [1951]) is mediated by plasmids present in bacterial populations (for a review see Davison [1999]). They are of benefit to a bacterial population due to their role in; expressing resistance to rarely occurring toxins and in providing protection from environmental stresses. This is a biological system that comprises two mutualistic populations; where one population (in this case the plasmid population) is *dispersed* amongst another population (the bacterial population).

It is examples such as these that imply the need for models of multi-level selection, since the survival of individual bacteria relies significantly upon a features of the population as a whole. Under the standard model, one might expect individual bacteria to evolve with a degree of resistance to hosting plasmids (for which there is a metabolic cost). However, in so doing, such a bacteria would be less likely to take advantage of benefits denoted by the inclusion of plasmid DNA. If we are to argue that the bacterial population in some sense acts as multi-cellular individual, then we still need a model that accounts for darwinian selection at the level of the individual, as well as that of the population.

This requires a further extension to the standard model (attempts to provide such a framework include Bresch et al. [1980], Boucher and James [1982] and Frank [1995]). Whilst these issues do not invalidate the core ideas of the modern synthesis, the need for an extended model remains an issue of contention (Pigliucci [2007], Brooks [2011]), potentially requiring a re-prioritization of lines of enquiry within biology (Lewontin [2002], Tauber [2010]).

A extension of the modern synthesis to incorporate multi-level selection may be of particular interest to evolutionary robotics researchers in the future. For ex-

ample in models of adaptation that incorporate a population of robots coevolving with a population of memes, in competition with other robot populations. Multi-level selection models, may also be of interest to those robotics researchers looking at cooperative activities (Baldassarre et al. [2003a]; Nitschke [2003]; Waibel et al. [2009]) or the evolution of communication (Yanco and Stein [1993], Steels [2003], Floreano et al. [2007], Greeff and Nolfi [2012]).

For the vast majority of research conducted in evolutionary robotics, the modern synthesis is largely sufficient as the basis for models that abstract evolutionary principles in algorithms (since most of the work involved in evolutionary robotics focuses on the behaviour of individual robots). However, as computational power increases, allowing for more complex simulations, models of multi-level selection will have an increasing relevance.

2.3.1.3 The Origin of Algorithms for Evolution

It is possible to identify four areas in which computational algorithms have been inspired by biological theories of evolution (dating back to the 1950's); artificial life, computational biology, optimization and artificial intelligence.

- **Artificial Life:** Nils Aall Barricelli is arguably the first to conduct anything resembling evolutionary simulations on a computer in experiments carried out at the Institute for Advanced Study at Princeton in 1953, 1954 and 1956 (Barricelli [1962]). He published his early work (Barricelli [1954] and Barricelli [1957]) in the Italian Journal Methodos whose primary topic was that of symbolic logic. His experiments involved an idealized two-dimensional space in the form of a grid inhabited by entities (each of which corresponded to a single number).

  The numeric value of an individual defines its phenotype; its movement (akin to that of a chess peice) in the grid and spawning characteristics (for a brief summary of this work see Fogel [2006a], for a wider historical context see Fogel [2006b]). Barricelli's work contributes less to the notion of a genetic algorithm, than it does the idea of artificial life. He did not seek to attain biological realism, rather he sought to abstract certain principles of

biological organisms and explore them through computer simulation (Barricelli [1962]), and as such bears a close relationship to modern day work in *artificial life*.

Barricelli was later to collaborate with Jon Reed, working at the Oslo University in Norway, in what is probably the first example of evolutionary strategies being used in machine learning. Their paper describes the simulation of an evolutionary algorithm for playing a simplified game of poker (Reed et al. [1967]).

- **Computational Biology:** Some of the earliest examples of work that we would recognize as resembling genetic or evolutionary algorithms, are those described in a series of papers produced by A.S. Fraser and J. S. F. Barker (working at the University of Sydney) between 1957 and 1960 (Fraser [1957a], Fraser [1957b], Barker [1958b], Barker [1958a], Fraser [1960b], Fraser [1960a]). One paper in particular describes what we might regard as the general form of an evolutionary algorithm (Fraser [1960b]) but for the failure to include a mutation operator;

  1. Extract, without replacement, a pair of parents at random from the given set of parents.
  2. Form a set of progeny from these parents.
  3. Determine the phenotypes of the progeny.
  4. Select potential parents from the set of progeny.
  5. Repeat (1)-(4) until all parents have produced the specified number of . progeny.
  6. Print out any required information.
  7. Repeat (1)-(6) using the selected progeny as parents.

  R.C. Lewontin was also to carry out simulations involving an evolutionary algorithm in experiments to model the evolution of various alternative morphisms in house mice (Lewontin and Dunn [1960]). These examples represent the earliest uses of evolutionary algorithms.

- **Optimization:** Bremermann wrote a paper in 1962 describing experiments applying evolutionary algorithms to numerical problems for the purposes of optimization (Bremermann [1962]), however according to his account, the results were not particularly impressive. Despite not being named as a co-author, Bremermann refers to a; 'Mr. Salaff, who has done all the programming' (page 101, Bremermann [1962]). The lack of implementation details in the paper might lead us to question the extent of Bremermann's personal contribution.

  Another German, Ingo Rechenberg carried out optimization work on wing design (Rechenberg [1965]), developing what he termed ***evolution strategies*** (together with Hans-Paul Schwefel), the subject of his doctoral thesis (Rechenberg [1971]). Rechenberg's work relied on the use of a mutation operator without recourse to recombination. A solution to a problem was be formulated in a fashion that allowed for random perturbation of solution parameters. If the modified solution was superior to that of the unmodified solution it would replace its *parent*, if not, then the original solution would undergo further mutation.

- **Artificial Intelligence:** The field that we now refer to as artificial intelligence was to originated largely from members of the cybernetics movement who were to focus on a computational approach to intelligence.

  The work of R. M. Friedberg (Friedberg [1958], Friedberg et al. [1959]), outlines an approach to producing computer programs through an automated process that from a modern observers perspective bears some resemblance to a clonal genetic algorithm. In this process a superior program would be selected from a population of programs, which had undergone a modification of certain program parameters in an iterative process. However, he makes no specific reference to this as an evolutionary process, and fails to draw an analogy with evolutionary systems.

  Perhaps more significant was the contribution made to this field by L.J. Fogel, A.J. Owens and M.J. Walsh (Fogel et al. [1966]), who coined the term ***evolutionary programming***. In their approach they would model a problem with a finite-state machine, and in a fashion similar to that adopted

by Rechenberg, would mutate the finite-state machine model, replacing the original solution with the perturbed solution if superior. Recombination played no role in this algorithm.

In 'Adaptation in Natural and Artificial Systems' (Holland [1975]) John Holland describes an algorithm operating on of a population of genes, employing recombination and mutation operators, in a formulation that he would termed ***genetic algorithms***. He also sought to explain why such algorithms work, through his notion of *schema theory*. In this theory he models a solution as a string of bits, and seeks to explain why through the use of genetic operators, those partial solutions (*schemata*) that contribute to a solution that is above-average (in comparison with the population fitness), will proliferate. John Holland's formulation incorporates all the features of an evolutionary algorithm (as described in section 2.2), as it is commonly understood from the modern perspective.

N.L. Cramer (Cramer [1985]) proposed an algorithm whereby a simple computer language, defining binary multiplication operators and their arguments, could be evolved using genetic algorithms, to change the function of a computer program, rather than simply its parameters. This was to extended and generalized by John Koza (Koza [1989], Koza [1990]), referring to his approach as ***genetic programming***. The essential idea is to formulate a solution to a problem in a tree structures. In the tree, nodes are functions and terminal nodes are function arguments. It is through the use of recombination and mutation operators that a population of such structures undergo adaptation, based on the fitness of the solution represented by the evaluation of the tree.

Here we have referred briefly to the origins of the terms; evolution strategies, evolutionary programming, genetic algorithms and genetic programming, collectively known as ***evolutionary computation*** (Eiben and Smith [2010]). A number of reviews exist that cover the topics mentioned in greater detail, including; an historical snapshot of genetic algorithms (De Jong [1985]), the emergence of evolution strategies (Beyer and Schwefel [2002]), genetic programming (Koza and Poli [2003], Fogel [2006b]) and the early developments evolutionary computation

(Fogel [2010b], Fogel [2010a]), genetic algorithms and evolution strategies (Dianati et al.), and nature-inspired algorithms more generally (Zang et al. [2010]).

2.3.1.4 Issues in Evolutionary Robotics

Section 2.2 provides us with an overview of the structure of an evolutionary algorithm in the most general terms. Section 2.3.1 starts will a similarly general approach in introducing evolutionary theory. In seeking to add more detail to these general introductions, we shall first discuss the idea of a *fitness landscape*, before going on to outline how this affects algorithm design in evolutionary robotics.

The notion of a fitness landscape was introduced by Sewall Wright (Wright [1932]). Using the terminology that we defined in section 2.2, the *set of genotypes* (all the possible *allele values* consistent with the structure defined by the *chromosome*) maps to a notional *set of phenotypes* via the *ontogenetic function*. In conjunction with the environment in which these phenotypes exist, we arrive at the idea of a *fitness landscape*, representing the space of all possible fitness values that might be attained from the *set of phenotypes*. An evolutionary algorithm explores the fitness landscape. It seeks to discover peaks in this fitness landscape corresponding to regions in genotype space that map to phenotypes with high fitness values.

A *recombination operator* results in a discontinuous jump in this genotype space, since it arbitrarily combines the *allele values* of disparate (prior to convergence) parents. A mutation operator typically only changes a smaller number of parameters (per iteration) via random perturbation. Depending upon the magnitude of these perturbations, a *mutation operator* therefore allows for a finer grained exploration of genotype space than that allowed for by the *recombination operator*. A *jagged* fitness landscape is one in which small variations in *allele value* result in a large variation of fitness in the associated phenotype. The problem with a *jagged* fitness landscape is that the potential benefit of a *mutation operator* is reduced, since it is unable to climb a gradual slope of fitness improvement through random perturbation.

If the parameter space that maps to high value fitness regions in the fitness landscape is small, and the fitness landscape is *jagged*, the ease with which high

fitness value *genotypes* can be discovered is significantly reduced. We become reliant upon the *recombination operator* to discover these narrow peaks by chance. If *genotypes* existent in the *population* lack diversity or the population converges too quickly, the chances of finding such peaks in the fitness landscape becomes very small. In contrast, a *smooth* fitness landscape is one in which *mutation operators* in particular, are able to discover peaks corresponding to high fitness-values. In a *smooth* fitness landscape, even if the space of genotype parameter space is again small, it is at least discoverable through the finer grained exploration that *mutation operators* allow. A *smooth* fitness landscape implies the existence of pathways through the fitness landscape, towards regions of higher fitness, that can be climbed through the random perturbation of *allele values*.

The notion of a fitness landscape, helps us to understand some of the core issues that affect design decisions in evolutionary robotics. It is not simply the functionality of the neural networks (that are a product of the ontogenetic function) that we need to consider, we also need to consider how *recombination* and *mutation operators* affect the efficiency with which are able to explore genotype space in the discovery of high-fitness individuals. What we shall do now, is to focus on a small number of narrowly defined topics within evolutionary theory that are particularly relevant to design decisions in evolutionary robotics; the Baldwin effect, *neutral* networks, modularity and diversity.

- **Baldwin effect:** The Baldwin effect provides us with a model for understanding the relationship between learning and evolution (Baldwin [1896]). Hinton and Nowlan conducted an influential experiment where they demonstrated that 'Learning alters the shape of the search space in which evolution operates and thereby provides good evolutionary paths towards sets of co-adapted alleles' (page 495, Hinton and Nowlan [1987]). This work was to become influential in the evolutionary robotics community, resulting in a resurgence of interest in the Baldwin effect (Mayley [1996], Harvey [1996], Nolfi [1997], Richards [2008], Lande [2009]). An interpretation of Baldwin's ideas from a modern perspective is that phenotypic plasticity results in a smoothing of the fitness landscape. The Baldwin effect requires two components; firstly, that *allele values* within a genotype allow for some variation in phenotype (plasticity), secondly, that there exists a feedback

process whereby the potential benefit of phenotype plasticity is exploited (e.g. if an organism can learn, and in so doing improve its fitness, there is a mechanism that ensures that this will take place). Under such circumstances, an organism in the course of its interaction with its environment is able (through the local search that its plasticity allows) to discover and engage in high-fitness activity, attainable within the limits afforded by this plasticity.

Baldwin's simple point was that a trait that is subject to adaptation during the life-time of the organism (a muscle strengthening under constant use, for example), denotes an advantage to the organism in that it allows it to take advantage of an opportunity that might not otherwise be subject to exploitation. Whilst there may be a metabolic cost associated with a more developed muscle, plasticity in the phenotype allows it to respond to environmental changes and competition pressures in a flexible fashion. The notion that learning endows an organism (or robot) with behavioural plasticity (rather than just physical plasticity) provides us with a model of how both learning and evolution interact in the pursuit of the evolution of robots that are capable of executing complex tasks. Chalmers was one of the first to demonstrate the evolution of connectionist networks capable of demonstrating learning (Chalmers [1990]). Urzelai and Floreana demonstrated the evolution of neural networks, whose dynamic properties were subject to Hebbian learning in real robot experiments (Floreano and Mondada [1996], Floreano and Urzelai [1998], Urzelai and Floreano [2000b]). Work conducted as part of this thesis also involved the comparative studies of such networks in the evolution of locomotion in robot simulations (McHale and Husbands [2004c], McHale and Husbands [2004b]).

- **Neutral Networks:** Motoo Kimura proposed an alternative mechanism to *natural selection* to explain variations within and between species based on a ***neutral*** theory of genetic evolution (Kimura [1983]) where variation is primarily the consequence of random process. The idea that stochastic variation might play a significant role in adaptation had been suggested previously (Maynard Smith [1970]) but in less detail than that presented by

Kimura. Under such circumstances, genetic diversity is created by variation that does not result in improved fitness (as required for natural selection under the Darwinian model). A diverse population arrived at through neutral variation (i.e. mutation that does not reduce fitness) acts as a substrate from which highly positive mutations might arise. This gives rise to the idea that the search space of phenotypic variation, might be efficiently explored via numerous pathways that exhibit no immediate increase in fitness, in other words; *neutral networks*. Where this idea is somewhat controversial (for a discussion see Leigh [2007]), is that whilst there is no reason under the Darwinian model that stochastic events resulting in genetic mutations that are neutral in their impact on fitness should be selected against, we cannot ignore the effects of stabilizing selection (Waddington [1959], Maynard-Smith [1983]), that results in a reduction in genetic diversity. The interest in neural networks from an evolutionary robotics perspective (Harvey and Thompson [1996]) relates to the idea that there may be benefits to devising algorithms where multiple pathways are available for evolutionary search, so as to improve the efficiency with which optimal solutions can be found (for more recent theoretical work see Barnett [2000] and Ebner et al. [2001]).

- **Modularity:** The notion of modularity in biological systems is different from the strict sense in which we might use it when referring to a physical mechanism. What is of particular interest to researchers in evolutionary robotics are the circumstances under which modularity can become a target for evolution (Wagner [1996]) and how this might influence decisions made in the evolution of neural networks (Happel and Murre [1994], Calabretta and Parisi [2005]). Ideally we would choose an encoding model and mapping procedure that would allow us to preserve positive traits (that might rely on some degree of functional modularity) so that we are able to explore a range of complex behaviours in our evolutionary algorithms. *Indirect encoding* (used in Cangelosi et al. [1994], Gruau [1994], Gruau and Quatramaran [1997], Eggenberger [1996], Eggenberger [1997b], Astor and Adami [2000], Bongard and Pfeifer [2003]) has the potential to preserve modular features. In such approaches, *ontogenetic functions* support mod-

ularity through some generative process, at a level which is not directly affected by the disruption of a *recombination operator* in the same way that *direct encoding* might be. The downside to approaches that are exclusively *indirect*, is that this might limit the exploration of intra-module adaptation. An example of an approach that combines both aspects of *direct* and *indirect encoding* is that used by Phil Husbands in his GasNet model (Husbands et al. [1998e]). In GasNets some *allele values* determine spatial sectors of connectivity (neuron nodes are deemed to be spatially distributed). Mixed encoding strategies formulations are potentially more robust to the preservation of functionality, than pure *direct* encoding strategies where alleles specify point-to-point neural connections, but yet still allow for the exploration of intra-module connectivity. A number of reviews have been written on the relative benefits of *direct* and *indirect* encoding (Cangelosi and Nolfi [2003], Kuscu and Thornton [1994]), Durr et al. [2010]).

- **Diversity:** One of the challenges of evolutionary algorithms is to try and maintain a degree of diversity in the population in an attempt to avoid premature convergence to a solution and allow for an efficient search of phenotype space (discovering novel behaviors). The maintenance of diversity in biology, is closely related to models of speciation (White [1968]), since speciation requires the long-term maintenance of diversity, ultimately leading to reproductive isolation (for a comprehensive treatment see Dieckmann et al. [2004] and Coyne and Orr [2004], for a review see Chesson [2000]). Speciation itself remains quite a controversial topic, with the origin of novelty and the processes by which diversity is maintained being key issues of debate (Baker [2005], Bolnick and Fitzpatrick [2007], Magnuson-Ford et al. [2010], Thibert-Plante and Hendry [2011]). Of particular interest is how reproductive quasi-isolation is maintained prior to adaptations that result in the biological impossibility of interbreeding. There are two processes by which diversity can be maintained; due to varying degrees of geographical isolation (Bolnick and Fitzpatrick [2007]), or through assortative mating (Ritchie [2007]). The role of a developmental mutation is of particular interest as far as novelty is concerned, as a small mutation is capable of

resulting in significant changes in phenotype.

Evolutionary algorithms make use of analogues of geographic isolation to maintain diversity through the adoption of population models that incorporate the notion of spatial distribution. In some cases a *tournament set* are selected from such a *population*, comprising the random selection of genotypes that are spatially close to one another. This ensures that competition is local, and allows for the preservation of dissimilar genotypes for longer than would be the case of no limitations were placed in gene flow. This is an analogue of parapatric speciation. This method was adopted in experiments reported in this thesis (McHale and Husbands [2004b], McHale and Husbands [2004c]). An alternative approach involves the reduction of gene flow between robots that have different behavioural attributes, which the authors term *behavioural speciation* (Trujillo et al. [2011]). Although not explicitly recognized by the authors, this is an analogue of assortative mate selection based on behavioural similarity. Other researchers jettison biological plausibility completely and take the short-cut of modifying (or replacing) the fitness metric by some measure of novelty (Risi et al. [2010], Lehman and Stanley [2011a]).

2.3.2 Neural Networks

The evolutionary robotics perspective differs from the connectionist perspective (for comprehensive texts on connectionism see Rojas [1996] and Haykin [1999]), in that it draws quite heavily on findings in neurobiology. As such some of the models used in evolutionary robotics can be closer to those used in computational neuroscience (which seeks to model real neural processes) than to those conventionally associated with connectionism (which seeks to create simple abstractions that are computationally efficient). Examples of this include approaches in evolutionary robotics inspired by neuromodulaton (Husbands et al. [1998a], Ishiguro et al. [2003b]), developmental processes (Gruau [1994], Eggenberger [1997b], Rust et al. [2001]) and dynamic plasticity (e.g. Urzelai and Floreano [2000b]).

Connectionism has traditionally focused on issues that relate to pattern recognition (Bishop [2006]), function approximation (back-propagation models, Rumel-

hart and McClelland [1986]), associative memory (Hopfield [1982]) and dimensional reduction (e.g. self-organizing feature maps, Kohonen [1990]). Evolutionary robotics makes use of these models where appropriate, but it has historically focussed on sensor-motor control systems (typically involving formulations of neural networks that were appropriate for continuous or discrete time simulation). The goal in evolutionary robotics has been to seek to create entire agents capable of exhibiting minimal behaviour (with a view to supporting complex behaviour), rather than to address specific problem domains such as computer vision or natural speech recognition for example. The philosophy that underlies evolutionary robotics is based on that of embedded cognition, and takes the viewpoint that the agent is in a dynamic coupling with its environment. Connectionism does not prioritize the agent oriented perspective.

In seeking to understand how evolutionary robotics has arisen from its technological and theoretical antecedents, we should first look at the roots of connectionism; cybernetic, biological and ethological. There are a number of historical reviews that emphasize different aspects of connectionism; orthodox (Medler [1998]), philosophical (Aizawa [1992]), and functional (Valentine [1989], Cowan [1990], Bechtel and Abrahamsen [1991], Cowan and Sharp [2012]). Cordeschi provides us with a rare insight into early connectionist machines and compares them with those developed more recently (Cordeschi [2000]). Other authors relate connectionism to other fields, such as neuroscience and AI (Schwartz [1988]), cognitive science (McCloskey [1991]), and psychology (Rowe [1995]). Of particular interest to us in seeking to explain the anastomosus of evolutionary robotics, are the historical origins of the diverse conceptuals streams that have combined to result in this field of research. Aizawa seeks to do this by identifying three strains of research;

> 'This might be divided roughly into: (1) the mathematical/computational strain of connectionism following up the work begun by Rashevsky, McCulloch, and Pitts; (2) the neuropsychological work by Hebb and his colleagues; and (3) the more purely neuroscientific work by, for example, Sir John Eccles' (page 305, Aizawa [1992]).

We prefer a slightly different categorical framework. In order of roughly in-

creasing abstraction from biological systems, we can identify four streams of research;

1. **Neurobiogical**: Sir John Eccles was representative of a group of researchers who sought to understand the biology of neural systems (Eccles [1953]). As such, the preoccupation of these researchers was with the electrical and chemical processes that might underlie the biological function of neural systems. The goal was not to build abstractions or models of these systems, but to understand their operation in terms of more fundamental physical principles.

2. **Computational Neuroscience**: Taking Rashevsky's work as an example (Rashevsky [1935]), it is clear that his goals are to seek a mathematical basis for an understanding of the workings of the brain. Whilst his work predates the advent of the modern computer, this perspective is very much that of modern computational neuroscience. He seeks to use mathematics to arrive at models of biological systems that underpin the workings of the brain, writing papers (such as Rashevsky [1933]) that are similar to modern papers in computational neuroscience (other than the fact that the graphs are hand drawn, rather than computer generated).

3. **Neuro-ethological**: Hebb was clearly motivated by a desire to understand and model the neurological basis of behaviour (Hebb [1949]). Evidence of this interest in behaviour is indicated by other published work, that addresses 'Heredity and Environment in Mammalian Behaviour' (Hebb [1953]). Whilst Hebb's work, sought to be biologically realistic, his focus was on the behaviours and cognitive processes that simple models of neurobiology might help us explain. What we now call Hebbian learning, is a simple principle, that whilst biologically plausible, does not typically require the complexity of modelling that Rashevsky's approach employs.

4. **Cybernetic**: The focus of the cybernetic movement (covered in more detail in section 2.3.3) is on the underlying principles behind the mechanisms and processes of control. The movement included those that had in interest in building machines that might have some practical application. The

early cybernetics movement had within it two themes, that typified by the William Ross Ashby's holistic perspective, and the computationalist theme typified by the work of Alan Turing. The role that McCulloch and Pitts would play within this movement, was to provide support to the computational clique, with a view of the brain as a computer (for a detailed analysis see Michael and Arbib [2000]). They did so through work that demonstrated neural nets as capable of supporting logical operations (McCulloch and Pitts [1943]). The notion that the brain works as if it were a computer might appear somewhat surprising from a modern perspective, however that this was their view is unambiguous; 'What we have said so far insures that those neurons in a brain can be and are used as relays in a computer to gate all-or-none impulses' (page 190, McCulloch [1957]), and referring to a conversation with Norebet Weiner in the early 1940's 'He talked also of various kinds of computation and was happy with my notion of brains as, to a first guess, digital computers, with the possibility that it was the temporal succession of impulses that might constitute the signal proper' (page 10, Mcculloch [1974]).

In contrast with other streams of research, their model represents a very much simplified model of neurons and connections than that attempted by Rashevsky for example. In this and extensions of this work, the functional significance of these networks is deemed of more greater interest than theri biological verisimilitude.

Although the notion of the brain as a computer is not given much credence, the idea that the brain is in some sense is engaged in *computation* is something that has persisted within the connectionist movement in the machine learning community. Frank Rosenblatt introduced the *perceptron* (Rosenblatt [1958b]) neural network model, showing that a collection of perceptrons can act as a pattern detector or linear discriminator. One way of abstracting Hebb's idea is through the notion of a *weight*. A weight is a single numeric value that can be perceived as reflecting the strength of a connection between two neurons. We can imagine a single layer of perceptrons connected to an input and output stage. The input stage corresponds to a set of notional sensors, whose activation at any point

in time reflects some sensory state. These inputs are connected to the perceptrons, such that the activation of the perceptron is a function of the product of the each input values and the weight of the connection between this input unit and a specific perceptron. The output stage would, through selecting the perceptron with the highest value for a given sensor input be able to differentiate between a continuous range of sensor values, acting as a linear discriminator.

It is the value of the weights of connections between artificial neurons representing an input pattern to the perceptrons that determine the output of the preceptrons for a given pattern. Rosenblatt went on to suggest that the value of these weights provided the basis for the incorporation of information in the brain, supporting recognition and determining behaviour (Rosenblatt [1958a]). The idea that brain activity might in some way be facilitated by the adaptation of individual neurons to sensor inputs, and behaviour result from the functional characteristics of individual neurons in collective activity is at the heart of connectionism.

Unfortunately this line of enquiry was inhibited by the publication of a book by Minksy and Pappert (Minsky and Papert [1969]) demonstrating the limitations of the perceptron, namely, its inability to solve the XOR problem (in boolean logic and a TRUE value occurs where both inputs are either true or false). The reason for this is that a single layer perceptron is a linear discriminator, and the discriminator boundary (a function of weights) between exemplars required to appropriately classify them needs to be curved for the XOR problem. Whilst this problem can be solved through the use of a two layer perceptron, this arguably was not known, or was not publicized widely at the time of Minsky and Pappert's book. That Minksy and Pappert's work was a deliberate attempt at derailing the connectionist agenda in the competition for funding is effectively admitted by Pappert 30 years later; 'By 1969, the date of the publication of Perceptrons, AI was not operating in an ivory tower. Money was at stake' (page 7, Papert [1988]). For as long as neural networks provided support for the idea of the brain as computer, it was of political value to those engaged in research involving computation. As soon as it became apparent, that models of neural networks might in some way present an alternative basis for addressing issues of artificial intelligence, then competition for funding was to became an issue. That neural

networks offered an alternative was made explicit in the paper 'Computation in Neural Nets' (von Foesrter [1967]). It is perhaps also significant that Ross Ashby's comments were elicited and acknowledged by the author of this paper, giving support to the notion of neural nets as an alternative to approaches favoured by the computational clique.

Although Paul Werbos (Werbos [1974]) proposed a method of weight adjustment in multi-layer perecptron networks that would allow perceptrons to solve the XOR problem, it was not until almost a decade later that interest in this area returned. This was largely due to work carried out by Rumelhart and McClelland (Rumelhart and McClelland [1986]) and their letter published in Nature entitled 'Learning representations by back-propagating errors' (Rumelhart et al. [1986]). Earlier papers outlining details of the Hopfield Network (Hopfield [1982]) for associate memory and the Self-Organizing Feature Map (SOFM) or Kohonen Network (Kohonen [1982], Kohonen [1990]) also contributed to a resurgence of interest in artificial neural networks. Despite this resurgence, the field would continue to undergo attacks from the computationalist oommunity. Papert for example continued to criticize 'misleading attempts to use connectionist methods in practical applications' (page 5, Papert [1988]). In the same paper he goes on to dismiss the (at that time) recent excitement over neural networks to be largely down to cultural dynamics; what he characterizes as an attraction to holistic thinking rather than rationalism. Connectionists would also see their work challenged by eminent biologists such as Francis Crick on the grounds of biologically implausibility. He was sceptical of long-term prospects of this field (Crick [1989]). The connectionist approach was also rejected by some psychologists, based on the argument that the brain was not connectionist at the cognitive level (Fodor and Pylyshyn [1988a]). Rodney Brooks was also to falsely characterize connectionism as primarily a solution to the problem of *representations*, an approach which he deemed flawed in seeking to address issues of artificial intelligence (Brooks [1991]).

Despite these objections, connectionist models, currently play a fundamental role in evolutionary robotics (see section 2.3.5.3). History will be the judge of Papert's assertion that connectionism has no role to play in practical applications.

2.3.3 Cybernetics

The origin of the word *cybernetics* is taken taken from the greek *kybernetes*, meaning steersman (Wiener [1948]). Norbert Weiner, the author of 'Cybernetics or Control and Communication in the Animal and the Machine' (Wiener [1949]) and one of the founders of the cybernetics movement defined cybernetics as follows;

> Cybernetics is a word invented to define a new field in science. It combines under one heading the study of what in a human context is sometimes loosely described as thinking and in engineering is known as control and communication. In other words, cybernetics attempts to find the common elements in the functioning of automatic machines and of the human nervous system, and to develop a theory which will cover the entire field of control and communication in machines and in living organisms. (page 14, Wiener [1948])

Ross Ashby, another founder of this field wrote 'Introduction to Cybernetics' (Ashby [1956]), and 'Design for a Brain' (Ashby [1960]). He characterizes these works himself in the following fashion;

> I. to C. is concerned with first principles, as they concern the topics of mechanism, communication, and regulation; but it is concerned with the principles and does not appreciably develop their applications. It considers mechanisms as if they go in small discrete steps, a supposition that makes their logical properties very easy to understand. Design for a Brain, while based on the same principles, mentions them only so far as is necessary for their application to the particular problem of the origin of adaptive behaviour. It considers mechanisms that change continuously (i.e. as the steps shrink to zero), for this supposition makes their practical properties more evident. (p vii,Ashby [1960])

Both scientists were to make references to we see references to *stability* and *feedback* (ideas central to modern control theory), information theory (Shannon

[1948]), and biological regulation. MucCulloch, presents a personal perspective (as a member of the early cybernetics movement) (Mcculloch [1974]), whereas Phil Husbands (Husbands et al. [2008]) puts the movement in a broader historical context.

Whilst the field was initially united in its underlying focus on the mechanisms and processes of control, there are two separate strains of thought that ultimately would lead to a tortoise and hare like competition. The first strain, that of the field of computation, is essentially work that is a derivative of the notion of the *Turing Machine* and its extension in the *Church-Turing* thesis. This essentially **mechanistic** approach to control accelerated quickly from its early origins, giving rise to the field of artificial intelligence (for a historical perspective see Kline [2011]).

In contrast, the second strain developed with an emphasis on an **analogue** approach to control (where analogue is used in the sense of continuous variables). Ross Ashby's work on adaptive systems (Ashby [1960]) sought to determine on general principles of control and regulation inspired by those discerned in organisms (for an overview see Umpleby [2009]). His models of adaptive processes were intended to reflect general principles that were equally applicable to artificial devices. He was to construct a physical implementation of these ideas in his Homeostat;

> (We can describe it either as 'a machine to do our thinking for us' or, more respectably, as 'an analogue computer') One was built and called the 'Homeostat'. (p99, Ashby [1960])

Simple robot agents were a theme of W. Grey Walter's work; his *turtles* were capable of autonomous motion, with brains of analogue electronics. They exhibited rudimentary photo-taxis (Walter [1950]) and primitive learning (Walter [1951]). This work is complementary to that of Ashby, in that it demonstrated that primitive behaviours could be supported by analogue systems, without the need for programming.

Although labelling these two strains as *mechanistic* and *analogue* understates the complexity of the ideas encompassed by these two perspectives, they are convenient labels. The two approaches were to take divergent paths. This is despite

their common interest in understanding the principles of; control, regulation and adaptive behaviour, so that these principles might also be applied to the construction of artificial devices. Although the *analogue* line of enquiry was to fade in comparison with that of the *mechanistic* line of enquiry, it has recently prospered in the somewhat related form of *nouvelle AI*. Understanding both these paths of enquiry is necessary for a complete understanding of the underlying concepts inherent in evolutionary robotics, although they differ in substantially in their contributions. The *mechanistic* approach is significant in that it represents in some sense the anti-thesis of ideas inherent in the *analogue* approach (as exemplified by the work of Ashby in particular). However, it is through the perceived limitations of the *mechanistic* approach, that alternative approaches have come to be re-defined. It is as one potential solution to these limitations that has resulted in the emergence of evolutionary robotics.

The foundations of this *mechanistic* approach were laid by Alonzo Church and Alan Turing (Church and Turing [1937]). They proposed what would subsequently be referred to as the *Turing Machine* concept, where a hypothetical device capable of manipulating symbols on a strip of tape according to a table of rules, is able act as an automatic computing machine. The Church-Turing thesis (although not formally proven) is that everything that is algorithmically calculable is calculable by a Turing Machine.

In 1945 Turing went on to design one of the worlds first computers. In his book, 'The Essential Turing', Jack Copeland provides us with description of this historical event (see Chapter 9 of Turing [2004]), together with a reproduction of Turing's 1947 *Lecture on the Automatic Computing Engine*. Cyberneticians Walter Pitts and Warren McCulloch (McCulloch and Pitts [1943]) were to consider how the human brain might support logical calculus through neural activity (see Moreno-Díaz and Moreno-Díaz [2007] for a perspective on McCulloch's historical legacy). Piccinini (Piccinini [2004]) refers to McCulloch's work as the first computational theory of mind and brain.

In 1945 John Von Neumann was also to devise a computer in a form subsequently referred to as the Von Neumann Architecture (von Neumann [1993]). This laid the foundation for modern computers; a CPU containing registers (including a program counter), memory to store instruction code and data, as well

as support for mass storage together with input and output mechanisms. Whilst Turing was aware of Von Neumann's rather abstract paper, Turing's own design was less complex (Copeland and Proudfoot [1996]) and was detailed to the degree that the electronic circuits required to implement his design were specified. This computer he would call the Automatic Computing Engine (ACE).

Alan Turing was one of the principal founders of modern computing and gave some consideration to the notion of computational intelligence (Turing [1950]). Despite this, based on the content of private communications with Ashby (Turring [1946]), it is clear that he had some appreciation of how the computational approach might differ from what Ashby was seeking to achieve. He states;

> In working on the ACE I am more interested in the possibility of producing models of the action of the brain than in the practical applications to computing.

Turing addresses criticisms (presumably made by Ashby) that such a system would be entirely uncritical when something goes wrong, and that it would be devoid of originality, replying;

> There is, however, no reason why the machine should always be used in such a manner: there is nothing in its construction which obliges us to do so. It would be quite possible for the machine to try out variations of behaviour and accept or reject them in the manner you describe and I have been hoping to make the machine do this. This is possible because, without altering the design of the machine itself, it can, in theory at any rate, be used as a model of any other machine, by making it remember a suitable set of instructions.

Ashby was not to accept Turing's request for collaboration, seeking instead to the create the *homeostat*, an electronic device documented in his 'Design for a Brain' (Ashby [1960]). As described by Peter Cariani (Cariani [2009]), the homeostat was intended as the embodiment of adaptive control. We shall revisit Ashby's work in more detail in section 2.5.2.1.

The exchange between Ashby and Turing is indicative of a methodological schism in the cybernetics movement. The Ashby perspective focusses on the

system, and the underlying dynamics from which the coherency of the system emerges. The dynamics of the system ensure a coupling of the agent with its environment, such that an external event might engender a destabilization of the system (eliciting a response, based on the interaction of potentially conflicting sub-systems). Subsequent to this disturbance, the system would return either to a neutral state, or an appropriate dynamic state. In his first edition of 'Design for a Brain', Ashby used the term *ultrastable* to apply to a specific system comprising two feedback loops, and *multistable* to apply to a variant incorporating more complex connections. However by the second edition (Ashby [1960]) he would use the term *ultrastable* to refer to the class of systems, incorporating everything from his original concept to that of systems comprising an arbitrary arrangement of such coupled sub-systems. The terms are therefore used somewhat interchangeably, as in;

> In other words, within a multistable system, subsystem adapts to subsystem in exactly the same way as animal adapts to environment. (p210, Ashby [1960])

In other places the term *ultrastable* is used in the same sense;

> Exactly the same principle governs the interactions between three subsystems. If the three are in continuous interaction, they form a single ultrastable system which will have the usual properties. (p210, Ashby [1960])

To differentiate between the two, it is perhaps convenient to refer to the dynamic underlying these system changes as that of *multistability*, engendered by the interaction of multiple *ultrastable* sub-systems.

Ashby seems resistant to Turings overtures, based on his suspicion that a computational approach might be over-deterministic, lack spontaneity or originality, but most importantly fail to embody the necessary organizing principle that might resolve competing computational instructions, that is at the centre of Ashby's ideas.

Ultimately this schism would become a chasm, as one of its branches outgrew the others; namely, the ideas which related to the theory of computation and its

derivatives. It is trivially simple to understand why this might be the case. If the central issue was control, then this can be simply achieved by the construction of a machine capable of executing instructions.

The problem with Ashby's approach was evidenced in the his construction of the *homeostat*. Training was achieved by the *punishment* (p110, Ashby [1960]) of the system, by an the introduction of an extreme disturbance that would result over time in the adaptation of the system to the preferred response, at which point the punishment would no longer be applied. In comparison with the computational alternative, this is clearly far less direct. It also relies upon the construction and connection of a machine which incorporates the necessary sub-systems, capable of supporting an appropriate response, without explaining any principle as to how this might be achieved. As will become evident later in this section, it is as a potential solution to this problem that evolutionary robotics has emerged. Artificial neural networks support the kind of dynamic interaction that Ashby envisaged in the construction of his *homeostat*. Employing evolutionary algorithms help us to explore the space of possible designs that might be receptive to adaptation to external influences. Prior to the emergence of this field however, the computational approach appeared to offer the simplest approach to achieving the goals of the early cybernetics movement.

2.3.4 From AI to Embedded Cognition

2.3.4.1 Artificial Intelligence

The advent of Artificial Intelligence is commonly deemed to start from a conference held at Dartmouth College in the summer of 1956 (for the conference proposal see McCarthy et al. [1955]). Whilst both *automatic Computers* and *neuron nets* were on the agenda, it was subsequent successes in the former that attracted funding. A *General Problem Solver* computer program was written as early as 1959 (Newell et al. [1959]). Shortly after this Minsky's paper 'Steps toward artificial intelligence' (Minsky [1961]) provides us with snapshot of the issues that were deemed central to artificial intelligence research, the titles of key sections appear as follows;

1. The problem of search

2. The problem of pattern recognition
3. Learning Systems (success-reinforced decision models)
4. Problem Solving and Planning
5. Induction and Models

These issues were deemed solvable through developing mappings between the real world and appropriate representations, and through the creation of algorithms that could support the symbolic manipulation of these representations. This emphasis on representation, symbolic manipulation, a focus on algorithms and data structures (and a complete lack of requirement for embodiment), is now referred to as *GOFAI* (Good Old Fashioned AI), a term coined by John Haugeland (Haugeland [1985]), or alternatively *classical* or *traditional AI*, and depending on context, *machine learning*. A review of the accomplishments of AI up to 1970 includes (Siklssy [1970]); axiomatic logic (1956), checkers (1959), propositional logic (1959), plane geometry (1959), calculus (1961), geometric analogies (1963), algebra word problems (1964), speech analysis (1966), speech synthesis (1968), hand-eye coordination (1968), and a seeing program (1968). Although in many cases the achievements referred to remain first attempts, they are an indication of the variety of problems that were addressed through these GOFAI methods, and a gradual change from tasks amenable to symbolic interaction, to ones that were more applied (such as speech, vision and robotic control).

The confidence of researchers in the agenda described my Minsky around this time, is indicated in the title of Newell and Simon's paper: 'GPS, A program that Simulates Human Thought' (Newell and Simon [1963]). However, it was not soon afterwards that AI researchers would experience their first setback, as doubt in the effectiveness of this approach to machine translation was questioned by a US government appointed committee (Seitz [1966]), leading to a decline in funding for this area of research for a period of approximately 10 years (Hutchins [2003]). From an entirely philosophical perspective Hubert L. Dreyfus, rather presciently, felt that this entire research agenda was fundamentally flawed (Dreyfus [1967]), arguing that intelligence requires embodiment. The UK government at the start of the 1970's also came to a critical conclusion of the field, in what came to be

known as the *Lighthill report* (Lighthill [1973]). Patrick Hayes (Hayes [1976]) made the point that general systems theory rather than computational intelligence best fitted our understanding of brain science, one of the few critics of the computational agenda from the academic community. Despite these criticisms, it was hoped that the ever increasing performance of computer hardware (Moore [1965], for a more modern perspective see Dongarra [2006]) would underpin future successes in the computational approach to intelligence, as remains the case to this day (Deb [2008]).

2.3.4.2 The Schism

A major schism in the AI community took place in 1990, when Rodney Brooks criticized the last 30 years of mainstream efforts in Artificial Intelligence (Brooks [1990]). He claimed that the central problem was that AI was not *physically grounded*, and proposed a new software architecture that relied heavily upon the physical embodiment of an intelligent agent. A position that is reminiscent of the ideas propounded by Hubert L. Dreyfus more than 20 years earlier (Dreyfus [1967]). Brooks sought to address these issues with his *subsumption architecture*, eschewing centralized control, relying upon the hierarchical resolution of competing reactive units. From a cybernetic perspective, one cannot but help see this as a computational analog of Ashby's notion of *ultrastability*, which others have termed using a more modern vocabulary *adaptive control* (Cariani [2009]). Brooks went further in criticising the core ideas of the computationalist agenda with a paper attacking the notion that symbolic representation was required by an agent exhibiting (from the distal observers perspective) intelligent behaviour;

> It is only the observer of the Creature who imputes a central representation or central control. The Creature itself has none; it is a collection of competing behaviors. Out of the local chaos of their interactions there emerges in the eye of the observer, a coherent pattern of behavior (page 148, Brooks [1991]).

Catalysed by Brooks' work, the 1990's saw the start of a period where the computationalist agenda was increasingly questioned. In sympathy with Brooks'

position, Willshaw considers 'Non-Symbolic Approaches to Artificial Intelligence and the Mind' (Willshaw et al. [1994]). Advocates of mainstream AI also saw the need for soul-searching. Simon's paper is interesting in that it gives us an insight into the mindset of a classical AI researchers around this time;

> Robotics is also a vigorous field, and also in danger of being too far separated from mainstream AI. I viewed with mixed feelings the establishment of a separate graduate program in robotics in my own university. One of the common criticisms of much mainstream AI research is that no distinction is made, in modelling problem situations, between the actual, real-world, situation and the model of the situation stored in computer memory. As robotics cannot afford this luxury of confusing the model with external reality, it must incorporate in its systems feedback channels that can correct the models periodically to reflect reality more accurately. Of course, this distinction can be attained in AI modelling, by keeping in memory both an abstracted model and a simulated *real world*, but the virtue of robotics is that it makes the distinction a necessity instead of an option, and continually reminds the system builder of the complexity of the *real* real world - the one outside the computer (page 123, Simon [1995])

It is indicative of the classical AI mindset, that even within robotics Simon saw the need for an internal model of the world, rather than Brook's solution which was to see the world as the best model of itself. Brooks may not have been aware at the time that his work provided other researchers with the hammer that would be used to attack those of the computationalist persuasion. He denied the influences of Heidegger (whose work had informed Dreyfus' views on the need for physical embodiment, Dreyfus [1967]) on his work, and by implication failed to recognize that his work might merit a significant change in the philosophical underpinnings of AI. He saw the *subsumption architecture* merely as an engineering solution, to an engineering problem, in the section entitled 'It isn't German philosophy' he writes;

> In some circles much credence is given to Heidegger as one who understand the dynamics of existence. Our approach has certain sim-

> ilarities to work inspired by this German philosopher but our work was not so inspired. It is based purely on engineering considerations (page 155, Brooks [1991]).

In the preceding section he also states that 'It isn't connectionism';

> .. connectionists seem to be looking for explicit distributed representations to spontaneously arise from their networks. We harbor no such hopes because we believe representations are not necessary and appear only in the eye or mind of the observer (page 154, Brooks [1991]).

Whilst it is true that very few modern practitioners would recognise this depiction of connectionism, at the time the notion that connectionism supported a form of sub-symbolic representation, was seen as a *defence* of connectionism. This relates to a theoretical debate initiated by Fodor and Pylyshyn (Fodor and Pylyshyn [1988b]). Their challenge was 'to explain the existence of systematic relations among cognitive capacities without assuming that mental processes are causally sensitive to the constituent structure of mental representations' (abstract, Fodor and McLaughlin [1990]). Smolensky replied in defense of connectionism;

> In calling the traditional approach to cognitive modeling the *symbolic paradigm*, I intend to emphasize that in this approach, cognitive descriptions are built of entities that are symbols both in the semantic sense of referring to external objects and in the syntactic sense of being operated upon by symbol manipulation. These manipulations model fundamental psychological processes in this approach to cognitive modeling. The name *subsymbolic paradigm* is intended to suggest cognitive descriptions built up of entities that correspond to constituents of the symbols used in the symbolic paradigm; these fine-grained constituents could be called subsymbols, and they are the activities of individual processing units in connectionist networks (page 3, Smolensky [1988]).

It does rather appear as if Smolenksy had allowed himself to fall into an intellectual trap. Those who supported the computationalist agenda (strong on representation) chose to attack connectionism on this basis, whilst Smolensky sought to defend it with a quasi-representation argument. This is despite understanding that 'connectionist models considered are massively parallel numerical computational systems that are a kind of continuous dynamical system.' (abstract, Smolensky [1988]). It is perhaps only in understanding the debate that was raging at the time, that Brooks' comments appear to make sense.

By eschewing connectionism however, he was to miss out on major new developments that would emerge from an entirely new conceptual framework; *computational neuroethology*, 'the use of modeling and simulation to study adaptive behavior' (synopsis,Chiel and Beer [2008]). Both Randall Beer (Beer [1990]) and Dave Cliff (Cliff [1990]) were to combine ideas from; *neuroscience* (the study of nervous systems) and *ethology* (the study of animal behaviour), with the prefix *computational* indicative of the notion that the subject of study was to be that of autonomous or simulated artificial agents (Cliff [2002]). Whilst both Beer and Cliff sought to adopt the *world as model* perspective, their methodology was strongly connectionist. Their emphasis was on the coupling achieved between the robot and environment mediated by neural sensorimotor systems;

> It has been argued that the connectionist paradigm is biologically in vacuo and in this sense is no advance on the symbolic paradigm: connectionism has acted merely as a palliative for several of the maladies of symbolism. The solution to this problem is linking the model neural network to the external world via a sensorimotor system, thus grounding the symbols in the model. Such an approach is more in line with the philosophy of the Maturana school. Cliff [1990]

Brooks admitted some concerns over the potential limits to his architecture (Brooks [1991]) that would prove to be well founded. His principal concern was over the number of *levels* that their designs were able to support. Thus limiting the complexity of behavioural that they might be able to demonstrate. Intrinsic to the *subsumption architecture* is the notion of a hierarchy, where each higher level

corresponds to both a mediator and arbitrator of lower level modules. He also admitted that learning in his architecture had not been sufficiently addressed, and that difficulties of integration arose when seeking to incorporate existing equipment or software (due to the preconceptions that were inherent in their designs). One might have imagined that the idea of using genetic or evolutionary algorithms as a methodological approach would have appealed to Brooks. However there is little indication that he did so. Although Brooks used the term *evolve* in the title of his paper 'A robot that walks; emergent behaviours from a carefully evolved network' (Brooks [1989]), the sense in which he used it appears to be that of *change* rather than biological evolution. No details of the use of a genetic algorithm are described in his paper. It was not until later (Brooks [1992]) that Brooks speculated on the possible use of genetic programming to solve some of these programs, but notable is the lack of any reference to the possibility of evolving neural networks (despite early published work in this area, which we discuss in section 2.3.5.3). We can only speculate that this might be because this would be tantamount to acknowledging the failure of his subsumption architecture, given that it would represent not just a difference of methodology, but also undermine any notion of the need for an explicit subsumption architecture.

Brooks failed to see the opportunity for a new philosophical perspective on Artificial Intelligence. He also failed to consider connectionism as an alternative approach to resolving these limitations. His work remains however significant, not so much because of the new methodologies that he developed, but because he sought to question the central idea inherent in classical AI; that representation and symbolic manipulation are at the heart of artificial intelligence. His approached would later be referred to more generally as *behaviour based robotics*(Arkin [1998]).

2.3.4.3 Embodied Cognition

In the early 1990's we say the start of a dramatic shift in ideas relating to cognition, of such breadth, that it is not easy to identify the causal roots, nor to define succinctly the new terms associated with this change in perspective. What we are seeing is an example of something as close to Kuhn's notion of a paradigm shift (Kuhn [1996]) as can be imagined. Although ill-defined, the concept of *embodied*

cognition is central to this shift in perception. From an evolutionary robotics perspective, it is significant because it represents a radical shift in perspective from that of classical AI to nouvelle AI;

> Might it not be more fruitful to think of brains as controllers for embodied activity? That small shift in perspective has large implications for how we construct a science of the mind. It demands, in fact, a sweeping reform in our whole way of thinking about intelligent behavior. It requires us to abandon the idea (common since Descartes) of the mental as a realm distinct from the realm of the body; to abandon the idea of neat dividing lines between perception, cognition and action; to abandon the idea of an executive center where the brain carries out high-level reasoning; and most of all, to abandon research methods that artificially divorced thought from embodied action-taking.
>
> What emerges is nothing less that a new science of the mind: a science that, to be sure, builds on the fruits of three decades' cooperative research, but a science whose tools and models are surprisingly different- a cognitive science of the embodied mind. (page xiii, Clark [1997])

Surprisingly, many of the ideas implicit in notions of embodied cognition are not new. The agent oriented perspective, where consciousness is a function of intentionality (in the sense that consciousness is directed towards goals, and that the meaning or significance of objects is defined in the context of these goals) was promoted by Franz Brentano in 1874 (Brentano and McAlister [1995]). An extension of these ideas by Heidegger placed emphasis on the notion of *Dasein*, whereby it is the context of the agent in their environment, and the imputed meaning that the environment has for the existence of the agent that defines consciousness (Heidegger [1962]). The psychologist Jerome Bruner identified the value, need for and significance of objects (within a given cultural context) as the principal factors in the construction of cognitive processes (Bruner and Goodman [1947], Bruner and Postman [1948]). Another psychologist James J. Gibson discussed the notion of *affordances* (Gibson [1977]), whereby cognition involves

the direct perception of objects (within an ecological context) based on their usefulness, value and potential to the agent. What is common to these ideas is the notion of *enaction*, whereby it is through the process of interaction with the environment that the agent's construction of cognitive abilities takes place.

From a modern perspective, the filter through which these ideas are perceived is that of embodied cognition, and the constituent notions of *situatedness* and *embodiment*. If we are to identify the catalysts for this modern reconceptualization of cognition, then the most likely candidates are the works carried out by Rodney Brooks on his subsumption architecture (Brooks [1990], Brooks [1991]), and the rather more abstract contributions made by Varela, Thompson, and Rosch in their book 'The embodied mind: Cognitive science and human experience' (Varela et al. [1992]). A number of reviews have been written on this subject, both from a cognitive science perspective (Wilson [2002]) and robotics perspective (Ziemke [2005], Anderson [2003], Froese and Ziemke [2009]). What we shall do however, is to focus on the two flavours of embodied cognition, associated with the catalysts that we identified earlier, which for shorthand reasons, we shall term *Brooksian* and *Varelian*.

None of the words; *situatedness*, *embodiment*, *situated*, *embodied* appear in Brooks' seminal papers (Brooks [1990], Brooks [1991]). The word *embodiment* appears only once, but in the context of its general usage in english. A considerable proportion of these papers describe details of specific robotic implementations of his ideas. As mentioned previously section, he saw his principal contribution as the subsumption architecture itself, a specific program architecture as a solution to an to an engineering problem. Retrospectively, we have seen an attempt to by other researchers to conceptualize his approach, through the notions of *situatedness*, *embodiment*. In so doing, these notions have been of particular use in the robotics community, in characterizing an approach, that is antithetical to the computational approach to artificial intelligence.

Arkin uses the term behaviour-based robotics to identify this new approach, employing as key principles (page 26, Arkin [1998]);

- **Situatedness:** The robot us an entity situated and surrounded by the real world. It does not operate upon abstract representations of reality, but rather reality itself.

- **Embodiment:** A robot has a physical presence (a body). This spatial reality has consequences in its dynamic interactions with the world that cannot be simulated faithfully

An alternative but consistent description is implicit in Pfeifer's design principles of autonomous agents, he states (page 7, Pfeifer [1996]);

- **Principle 1c:** The agents must be **embodied**, i.e. they must be realized as a physical system capable of acting in the real world. Although simulation studies can be extremely helpful in designing agents, building them physically typically leads to surprising new insights. This point has been forcefully made by Brooks (1991). Physical realization often facilitates solutions which might seem hard if considered only in an information processing context. An agent existing only in simulation would not be complete.
- **Principle 1d:** The agents must be **situated**, i.e. the whole interaction with the environment must be controlled by the agent itself, i.e. the world must always be seen from the perspective of the agent. Moreover, the agent has to be able to bring in its own experience in dealing with the current situation.

The *Varelian* flavour of embodied cognition is that proposed by Varela;

> Let me explain what I mean by the word **embodied**, highlighting two main points: (a) that cognition depends on the kinds of experience that come from having a body with various sensorimotor capacities; and (b) that these individual sensorimotor capacities are themselves embedded in a more encompassing biological and cultural *context*(page 329, Varela [1995]).

Whilst the foundation of the Brooksian flavour is teleological, i.e. its conception relates to the construction of intelligent agents, the Varelian flavour has a philosophical and psychological slant not present in the Brooksian perspective. This is exemplified by his attempts to define what he terms an *enactive approach to cognition*;

> In a nutshell, the enactive approach to cognition consists of two key points: (a) that perception consists in perceptually guided action; and (b) that cognitive structures emerge from the recurrent sensori-motor patterns that enable action to be perceptually guided (page 329, Varela [1995]).

It is the Brooksian flavour of embodied cognition that is generally applicable to autonomous robotics. The Varelian flavour essentially subsumes the Brooksian flavour, but at the same time places particular emphasis on the means by which this embodiment and situatedness is intermediated; through perceptually guided action supported by a neural substrate.

We recount here ideas on the subject of embodied cognition, largely because it is part of the vocabulary of modern robotics, and because its emergence as a concept (ill-defined as it is) represents a significant shift in perspective from that of the disembodied view of intelligence represented by GOFAI. We shall avoid subsequent reference to the term for two reasons. Firstly, since it lacks a commonly accepted definition the usefulness of the term diminishes. Secondly, it makes more sense to focus on the effects or manifestation of embodied cognition, when we are seeking to determine how it might be synthesized. The notion of *adaptive behaviour*, allows us to discuss the kind of behaviours that we are seeking to create in artificial agents, encompassing as it does, all forms of adaptation be they genetic, onotogenetic or related to learning. The idea that such adaptive behaviours have arisen via an evolutionary process is far more parsimonious, than any reference to embodidness and situatedness (since all evolved organisms have bodies and are situated). The primary value of the term embodied cognition is (at least as far as we are concerned in this thesis) in its antithetical meaning to the ideas inherent in GOFAI.

2.3.4.4 From Autopoiesis to Evolution

Hubert L. Dreyfus criticized the computational approach to artificial intelligence as early as 1967, based on its failure to incorporate the ideas of Heidegger on *enaction* (Dreyfus [1967]). His retrospective of the failure of AI is therefore of particular interest (Dreyfus [2007]).

To paraphrase his arguments; he first states that traditional AI's response to the 'common sense' problem was to claim that more data storage was required. When this failed to achieve results due to a lack of a framework for determining appropriate data, the concept of a *frame* was proposed (supporting a context dependent formulation of relevant data and affordances). Drefyus then claims that problems arose due to the potential recursive nature of frames. Whilst Dreyfus approved of the grounding that Rodney Brook's robotics perspective allowed for, he feels that the approach taken was overly ambitious in the long-term claims that it made. Dreyfus' claims that Brook's ultimately failed, seems largely based on a criticism of his implementation (that of the *subsumption architecture*, which relies on converting stimulus input to reflex responses. Dreyfus identifies the dynamics perspective (where the agent is perceived as being coupled with the external world, and responses governed by neuro-dynamics) as the most appropriate basis for future work.

Perhaps somewhat surprisingly, Dreyfus fails to acknowledge the work of Maturana and Varela on *autopoiesis* (meaning *self-produced*) in this historical perspective. Central to the ideas of autopoiesis are notions of *circularity* and the *structural coupling* of the agent with its environment (which we shall describe in more detail shortly). For those who have some familiarity with Ross Ashby's work (see section 2.3.3), their ideas may not necessarily appear particularly novel, but that does not mean to say that they have not been influential.

It is difficult to say whether or not their work came to prominence *because* of the requirement for a new philosophical basis for understanding the relationship of the agent with its environment (in support of the new perspective on AI suggested by Brook's work), or simply because it was an 'idea whose time had come'. Certainly, the long gestation of these ideas dating back to the 1970's (Maturana [1970]) imply that it is the latter rather than the former. In either case, the publication of two books around this time (Maturana and Varela [1987] and Varela et al. [1991]) certainly had an impact on the robotics community, and perhaps more generally, in the shift in focus from disembodied to embodied cognition. Irrespective of the long-term value of the concept of *autopoiesis* and the philosophy that has arisen around it, its role was to add an air of legitimacy to the notion that there was something fundamental about the concept of embodied cognition,

that would ultimately allow us to re-address the issue of artificial intelligence. Even the idea of artificial intelligence was to be replaced by that of behavioural complexity, in line with the notion that *intelligence* is something that is perceived by the observer and is merely an distal observation of more fundamental processes that govern organism activity (Keijzer [1998]).

There are a number of ideas implicit in the notion of autopoiesis proposed by Maturana and Varela, and their thesis is extensive, since it relates to a theory of life and cognition. It is not necessary to accept their agenda in its entirety however, since within it there exist smaller digestible chunks, that we are able to borrow, in seeking to present a philosophical justification for the use of evolutionary robotics to achieve behavioural complexity in an artificial agent.

Although multiply and numerously defined and redefined by the authors, the idea of *autopoiesis* remains a foundational concept. This relates to the organism as a self-maintaining (and potentially self-regulating) entity, supported by homeostatic processes;

> Living systems as they exist on earth today are characterized by exergonic metabolism, growth and internal molecular reproduction, all organized in a closed causal circular process that allows for evolutionary change in the way the circularity is maintained, but not for the loss of the circularity itself. (page 1, Maturana [1970])

> This circular organization constitutes a homeostatic system whose function is to produce and maintain this very same circular organization by determining that the components that specify it be those whose synthesis or maintenance it secures. (page 2, Maturana [1970])

This concept of **circularity** persists in their ideas, and is expanded upon by Thamoson and Varela in what they refer to as *radical embodiment*. They define its dimensions as;

> The relationship between neural dynamics and conscious situated agents can be described in terms of the participation of neural processes in the cycles of operation that constitute the agents life. Three kinds of cycles need to be distinguished for higher primates:

1. cycles of organismic regulation of the entire body;
2. cycles of sensorimotor coupling between organism and environment;
3. cycles of intersubjective interaction, involving the recognition of the intentional meaning of actions and linguistic communication (in humans).

(page 2, Thompson and Varela [2001])

A second idea relates to the notion of the relationship that an entity has with its environment, described as a **structural coupling**;

> The relation between a living system and the medium in which it exists is a structural one in which living system and medium change together congruently as long as they remain in recurrent interactions. I have called this relation structural coupling, and I have shown that a living system flows in its living in the path of conservation of structural coupling with a the medium that makes this possible, until the living system dies. Living occurs in the path of structural changes that continuously result in the conservation of autopoiesis and adaptation or structural coupling.(page 24, Maturana [2002])

If we compare this with the conclusion made by Dreyfus in his criticism of classical AI (Dreyfus [2007]), it is difficult not to come to the conclusion, that their biological perspective might well provide us with a some clue as to how we might best achieve, what Dreyfus terms 'Heideggerian AI';

> There is, however, a big remaining problem. Merleau-Pontys and Freemans account of how we directly pick up significance and improve our sensitivity to relevance depends on our responding to what is significant for us given our needs, body size, ways of moving, and so forth, not to mention our personal and cultural self interpretation. If we cant make our brain model responsive to the significance in the environment as it shows up specifically for human beings, the project of developing an embedded and embodied Heideggerian AI cant get off the ground. Thus, to program Heideggerian AI, we would not

> only need a model of the brain functioning underlying coupled coping such as Freemans; we would also needand heres the ruba model of our particular way of being embedded and embodied such that what we experience is significant for us in the particular way that it is. (page 265,Dreyfus [2007])

If we were to re-interpret the ideas of Maturana and Varela, we might do this parsimoniously by saying that they seek to characterise the notion of an organism. The organism exists through the regulation of processes that seek to ensure its existence (not a circular argument, just a circular cycle). It is coupled with its environment, such that these regulatory processes in some sense take into account the environmental context of the organism. It is through the description and identification of an organism in its environment that we are able to define the contextual frame of reference Dreyfus' deems so important in his analysis. As such, it is this concept of *organism-environment* that presents us with one solution to the problem of Heideggerian AI. If this is the case, then the central challenge of AI is one of creating an artefact that embodies these principles, in a fashion that is relevant to its environment. Ultimately the reason for both Ashby's and Brooks' failure to create such artefacts, were essentially the same: they were failures of implementation.

Ashby created the homeostat, a system comprising a composite of independent homeostatic systems, where each sub-system was intended to map to some kind of behaviour. A disturbance to the inputs of the system would result in a re-arbitration of the appropriate behaviour through the dynamic competition of these subsystems. Such an approach sought to avoid the brittle response of computational systems, and also provide some basis for the emergence of properties that had not been predetermined by a designer. The problem Ashby had, was that he had no methodological framework by which he could ensure that the dynamics of system that he sought to create, were appropriate in their behavioural response to environmental stimuli. The homeostat supported arbitrary dynamic processes, rather than ones which had been shaped through a process of evolution (in the biological sense of the word). Organisms however, exhibit constructive behavioural dynamics through the as a consequence of the process of evolution. The autopoietic organism is a product of evolution.

Brook's implementation problems again were methodological. The were due to the *requirement* for the involvement of a human designer. Whilst his desire to avoid representational world modelling firmly planted the robot in a structural coupling (to use Maturana and Varela's terminology) with its environment, his subsumption architecture required that the competing processes (the mediators of this structural coupling) be first imagined by its human designers and then explicitly programmed. Whereas in classical AI, the problem was one of devising appropriate data representations and symbolic manipulators (conceived of by the programmer) with all its associated limitations, in Brooks' case, the problem was one of determining the appropriate reactive responses intermediated by an arbitrator (conceived by the programmer) that would support 'intelligent' behaviour.

How then might we be able to resolve this implementation problem? We wish to avoid the need for a human designer making decisions that pre-determine which constructive behaviours should be supported, and yet have an implementation that can be imbued which characteristics that we might deem functionally useful. The solution to this problem is almost trivially obvious. If an organism is capable of fulfilling the requirements identified by Dreyfus, and we wish to avoid the human role in design, we simply need to use a process that we know works without a designer, and that process is *evolution* itself. Clearly, we still need a substrate upon which an evolutionary process can operate. The methodology of evolving neural networks is one that allows us to solve the implementation problem that was to defeat both Ashby and Brooks.

As in the case of embodied cognition, we have sought here to introduce ideas on autopoiesis, largely because they are deemed to be part of the conceptual fabric of modern views on the synthesis of complex behaviour. Autopoesis as a characterization of the organism-environment coupling and embodied cognition as the basis for complex behaviour represent philosophical and cognitive frameworks that help us to understand what principles are relevant to the construction of an intelligent artificial organism, but they do *not* define a methodology by which such entities can be synthesized. Evolution provides us with a model of the process by which organisms have come to exhibit complex behaviour. Evolutionary robotics in turn provides us with a methodology that allows to create such systems through; (a) the application of evolutionary algorithms to (b) con-

nectionist models of neural systems (c) embodied in real or physically simulated agents, in a fashion that is entirely consistent with ideas on autopoiesis, embodied cognition and Hubert L. Dreyfus' ideas on Heideggerian AI.

2.3.5 Convergence

Simply put, the **conceptual** underpinnings of evolutionary robotics are evolutionary theory and the notions of embedded cognition, and the philosophical ideas relating to enaction. The **methodological** underpinnings are those of evolutionary algorithms, neural networks and real (or simulated) robots. The primacy of evolutionary theory and biological plausibility is a characteristic of evolutionary robotics, and represents a major divergence from machine learning approach to artificial intelligence. This methodological difference has as its central premise the idea that it is only through emulating the process of biological evolution, that we are likely to develop systems of the requisite complexity (with no known inherent limit) necessary to support intelligent behaviour. What we seek to do in this section is to describe the convergence of the ideas described up to this point that have resulted in the genesis of evolutionary robotics.

2.3.5.1 Evolving Neural Networks

Key in the emergence of evolutionary robotics was the realization that the genetic adaptation of artificial neural networks (now referred to as *neuroevolution*) represents a viable methodology for solving a range of problems. Some of the earliest examples of this work were carried out by W.B. Dress working at the Oak Ridge National Laboratory. He presented his work at the first 'IEEE international conference on neural networks' in 1987 (Dress [1987]) and with J.R. Knisley at the 'IEEE international conference on systems, man, and cybernetics' (Dress and Knisley [1987]) in the same year. They employed a *mutation operator* on a network description file and an *ontogenetic function* for network construction. Dress was to recognize three key features of this approach; the reduced requirement for an 'omnsicient' human designer, the potential for creating systems that were less brittle than those offered by the computational approach, and the negative aspect relating to the onerous computational overhead incurred in conducting such sim-

ulations (Dress [1987]). From this date, we saw a gradual increase in the number of papers published on this topic (e.g. Harp et al. [1989], Montana and Davis [1989], Whitley and Hanson [1989]). However, it was not until 1990 that we see such papers being more widely cited. Influential papers published at this time include; the use evolutionary programming techniques to train back-propagation networks (Fogel et al. [1990]), the use of genetic algorithms to discover weights and topologies in feed-forward networks (Whitley et al. [1990]), the discovery of learning rate, momentum and initial weight parameters for back-propagation networks (Belew et al. [1990], Schaffer et al. [1990]), and the evolution of parameters in a developmental model, used for the purposes of generating neural networks (Kitano [1990]).

In 1992 a workshop entitled 'COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks' took place in Baltimore, Maryland, sponsored by the IEEE Neural Networks Council. Papers presented included; a review of current techniques (Schaffer et al. [1992]), the use of a genetic algorithm for training *dynamic* artificial neural networks (Elias [1992]), a comparison of recombination operators (Hancock), the genetic synthesis of boolean networks using a developmental model (Gruau [1992]) and an approach to evolving both network structures and weights (Dasgupta and McGregor). Early reviews of evolutionary techniques applied to neural networks were written by Xin Yao in 1993 (Yao [1993]) and by I. Kuscu and C. Thornton at the University of Sussex (Kuscu and Thornton [1994]). A preliminary taxonomy of these techniques was written in 1995 (Balakrishan and Honavar [1995]), in the same year that a review of neural network genotypes was written (Nolfi and Parisi [1995b]).

2.3.5.2 Precursors to Evolutionary Robotics

Perhaps the first example of the use of evolutionary techniques with robotics was conducted by researchers at the Wayne State University, in Detroit (Kirby and Conrad [1986]). Google scholar (at the time of writing) indicates no more than 41 citations of this paper, which on closer inspection yield no names amongst the early founders of evolutionary robotics. It may be that this paper was largely overlooked by the robotics community, as its title ('Intraneuronal dynamics as

a substrate for evolutionary learning') displays little indication that the paper relates to a robot navigational task, nor is the choice of publication (Physica D: Nonlinear Phenomena) one which is traditionally associated with robotics or artificial intelligence. Their evolutionary algorithm is intended to model the dynamic adaptation of *enzymatic neurons* described in Condrad's earlier work (Conrad [1974], Kampfner and Conrad [1983]). Their idea of neuronal adaptation is analogous to that of *experiential selection* described in Gerald Edleman's *neural darwinism* hypothesis (Edelman [1987]).

Conceptually, this work is quite sophisticated. Most mainstream approaches to evolutionary robotics focus on the process of *genetic adaptation*, as an analogue of Darwinian evolution. i.e. where the adaptation takes place as a consequence of sexual or asexual reproduction. The work conducted by Conrad, Kirby and Kampfner incorporates the idea that *dynamic* adaptation takes place in neural networks (i.e. *learning*) in a fashion that is analogous to Darwinian evolution. There is no notion of a population of artificial genomes from which the neural systems of robots are created, nor of the discrete adaptation that takes place at reproductive events. As a consequence, it represents an interesting model of learning in robots, rather than an early example of evolutionary robotics. Conrad in a later speculative work 'Towards an artificial brain' (Conrad et al. [1989]), describes an approach to building an artificial brain, which incorporates an evolutionary learning algorithm such that;

> The evolutionary learning algorithm may be interpreted either as representing the mechanism of variation and natural selection acting on a phylogenetic time scale, or as a conceivable ontogenetic adaptation mechanism. (abstract, Conrad et al. [1989]),

This idea anticipates developments that would take place in evolutionary robotics over the following years. Again, at the time of writing, this paper only elicits 52 citations on using the Google scholar website, with no citations from the early developers of evolutionary robotics. Conrad's work may well have been accidentally over-looked by the robotics community.

One of the earliest examples of the use of the genetic algorithm to support primitive motor controllers for simulated robots was work done by Hugo de Garis

in 1990. He reports the use of *genetic programming* (as distinct from the *neuroevolution* approach generally used in evolutionary robotics) to develop controllers for a pointer (a jointed kinematic structure) and a walker (a 2d simulated stick figure). Unfortunately the paper lacks details of the physics models used in these experiments, so it is difficult to assess the level of realism of these simulations, and therefore difficulty of the described tasks. He describes a two stage process where functional modules are first evolved independently, before they are conjoined and functional parameters or weights adjusted in the secondary stage (de Garis [1990]) of the evolutionary process.

Karl Simms was also to use an genetic programming approach initially in evolving two-dimensional images (Sims [1991]), but would go on to evolve three-dimensional physically simulated creatures capable of exhibiting primitive locomotion (Sims [1994]). Note that this later work occurred after the advent of evolutionary robotics, but was highly influential in stimulating future research in the use of evolutionary techniques to simulate virtual organisms. Simms work is one of the first examples of the use of evolutionary techniques for the generation of control systems *and* morphology.

2.3.5.3 The Birth of Evolutionary Robotics: 1992

Arguably, the year that might best correspond to that of the birth of evolutionary robotics is 1992. There appear to be three groups of researchers who it appears largely independently arrived at the methodology of evolving neural networks to simulate simple behaviour. Randall Beer and John Gallagher working at the Case Western Reserve University of Ohio published a paper recounting the evolution of controllers for supporting chemotaxis, and in a seperate experiment, locomotion in a hexapod (Beer and Gallagher [1992a]). Rather than a real robot, they used a very simple physics simulation where the velocity of movement is proportional to the force applied. Domenico Parisi, Stefano Nolfi, and Federico Cecconi working at the National Research Council in Rome (Parisi et al. [1992]) described the evolution of a network, used to support the behaviour of a simple creature living in a simulated grid world. The agent was evolved to accumulate food objects distributed in the grid. In a report written by Stefano Nolfi and Domenico

Parisi entitled 'Growing neural networks', a very simple developmental model was evolved for the purposes of food catching in the simplest of simulations (Nolfi and Parisi [1991]), this report was republished in (Nolfi and Parisi [1994]), from where it no doubt reached a wider audience.

Perhaps the most advanced work however was that carried out by Cliff, David Cliff, Inman Harvey and Phil Husbands at the University of Sussex in Brighton, where simulations of a wheeled robot with tactile sensors (Cliff et al. [1992]) and a visually guided robot (Husbands and Harvey [1992]) were conducted. It is from this group that we first see the term *Evolutionary Robotics* appearing (Harvey [1992a]). The same group of researchers extended their work considerably in 1993 (Cliff et al. [1993a], Cliff et al. [1993b], Harvey et al. [1993a]), continuing in their use of the term *evolutionary robotics*. Note that both Randall Beer (Beer [1990]) and Dave Cliff (Cliff [1990]) had both previously (and independently) coined the term computational neuroethology (as referred to in section 2.3.4.2) in reference to the notion of using neural networks as the basis for exploring the behaviour of artificial creatures in simulation. The Sussex group also made use initially of the dynamic neural network model proposed by Beer and Gallagher (Beer and Gallagher [1992a]).

Whilst the cole Polytechnique Fdrale de Lausanne (EPFL) would become a place of major work in evolutionary robotics, lead by Dario Floreano and Francesco Mondada commencing two years later (Floreano and Mondada [1994], Mondada and Floreano [1995]), in 1993 they were still doing work related to Rodney Brook's subsumption architecture (Mondada and Franzi [1993]). Other of work in Switzerland that was complementary to that evolutionary robotics movement took place at the University of Zurich under Rolf Pfeifer. Rolf Pfeifer was to concentrate on the design principles important to robots that were to operate under the notion of embedded cognition (Pfeifer [1996]), very much in keeping with the ideas of evolutionary robotics. It was not until much later however, that evolutionary robotics work was to be carried out at Zurich (Bongard and Pfeifer [2003]).

Following 1992 we would gradually see a significant increase in the use of these techniques applied to robots. This included; gait synthesis in a simulated hexapedal robot (Lewis et al. [1994]), a review of techniques to generate modular

networks, including evolutionary approaches (Happel and Murre [1994]), the evolution of modular networks generated via a developmental model in support of hexapedal locomotion (Gruau [1994]), a model of gene regulation to generate networks in support of foraging activity (Cangelosi and Elman [1995]), locomotion in a real hexapedal robot (Gallagher et al. [1996b]), cell interactions as a developmental model for control purposes (Eggenberger [1996],Eggenberger [1997b]), the evolution of morphologies (Eggenberger [1997a]), the evolution of Lindenmayer systems to generate networks in support of competitive organisms in a simulated environment (Channon and Damper [1997]), the simulation of locomotion in lampreys (Ijspeert et al. [1997]), gaits in quadrupedal locomotion Grasso and Recce [1999], locomotion in a miniature humanoid robot (Nordin and Nordahl [1999]) and the use of *incremental evolution* to generate a controller for a real hexapedal robot(Filliat and Kodjabachian [1999]). These represent just a sampling of early work carried out in evolutionary robotics. Since then a number of reviews and general overviews from differing perspectives have been written (Weiss [1993], Gomi and Griffith [1996], Harvey et al. [1997], Yao [1999], Floreano and Urzelai [2000b], Ruppin [2002], Nolfi and Floreano [2002], Pratihar [2003], Floreano et al. [2004a], Floreano et al. [2004b], Miconi [2007],Floreano et al. [2008], Floreano and Nolfi [2008], Nelson et al. [2009], Floreano and Keller [2010], Don [2011]).

2.4 Research in Evolutionary Robotics

There appear to be two broad groups of researchers interested in evolutionary robotics. Those based in Europe and those in the USA. A considerable proportion of the research output in Europe is concentrated in three main research centres, the University of Sussex at Brighton, the Federal Polytechnic of Lausanne (EPFL), and the National Research Council (CNR) in Rome (although some work has also been done at the University of Zurich, its primary focus is on autonomous robots rather than evolutionary robotics). In contrast, research in the USA (possibly with the exception of Hod Lipson based at Cornell) appears to be dominated by a dispersed group of individuals who have exhibited greater mobility than their European counterparts.

Whilst it is probably true to say that Evolutionary Robotics research was

quicker to start in Europe than it was in the USA, with two notably expections; that of Randall A. Beer initially at Case Western Reserve University and Jordan B. Pollack initially at The Ohio State University However since then growth of research in the USA has been significant. *Second generation* researchers such as Josh Bongard (now at Vermont University) and Keith Stanley (now at the University of Central Florida) and Hod Lipson (now at Cornell) have all started new research groups. Evolutionary Robotics research in the USA is now comparable with that in Europe, if it has not already surpassed it.

It is perhaps significant that two of the centres responsible for the emergence of evolutionary robotics in Europe were multi-disciplinary research groups. The researchers at Sussex in the Department of Informatics participate in cross-departmental research groups, including the Centre for Computational Neuroscience and Robotics (CCNR) and the Centre for Research in Cognitive Science (COGS). The researchers at the National Research Council are part of the Institute of Psychology, despite their leading work in robotics. One might speculate that the relatively slower start-up of research in the USA may be due to some form of institutional inertia related to the dominance of Artificial Intelligence research in the USA. Perhaps surprisingly, relatively little work has been done in Japan on evolutionary robotics. Once again, given Japan's historical commitment to robotics, one might surmise that this is also related to institutional inertia, given the relatively recent emergence of evolutionary robotics as a discipline.

This section is intended to give an overview of the work conducted by the principal research centres. The focus on institutions in Europe and individuals in the USA is not intended to diminish the work conducted by individuals at other institutions, but simply to highlight some of the key topics of research that are encompassed by certain individuals and institutions. It is intended as a representative selection of work that has been and is currently being carried out in the field, rather than a definitive review of all the work that has taken place in evolutionary robotics. Work by individuals at smaller research groups is under-represented although every attempt is made to mention relevant work in its appropriate place in this thesis.

Although, papers that relate to locomotion, robot controllers, minimal behaviours, neural network formulations and the interaction between learning and

evolution are of highlighted due to their relevance to this thesis, an attempt is made to outline alternative goals and themes that are of interest to other researchers in evolutionary robotics.

2.4.1 Europe

2.4.1.1 The University of Sussex, Brighton

It is probably fair to say that the researchers at the University of Sussex (David Cliff, Inman Harvey and Phil Husbands) contributed the significant bulk of early work in this area (Harvey [1992b], Cliff et al. [1992], Harvey [1992a], Cliff et al. [1993a], Harvey et al. [1993a], Cliff et al. [1993b], Harvey et al. [1993b], Harvey [1993b], Husbands et al. [1994], Husbands et al. [1995], Husbands et al. [1997], Cliff et al. [1997]). Of these researchers, Phil Husbands remains at the University of Sussex, Inman Harvey left in January 2011 (although remains a Visiting Senior Research Fellow in Informatics), and Dave Cliff left much earlier in 1997. A core focus at Sussex has been to seek to develop novel formulations of neural networks that are particularly suited to supporting primitive behaviours in wheeled robots with tactile sensors and primitive vision and locomotion in legged robots. A large proportion of this work employs the GasNet formulation of neural network, whose properties include a novel abstraction of neuromodulation. This work has involved the use of GasNets in robotics simulations, as well as in studies closer to those typical of computational neuroscience (Husbands et al. [1998e], Husbands [1998b], Jakobi et al. [1998], Husbands et al. [1998d], Smith et al. [2001b], Smith et al. [2001a], Husbands et al. [2001b]).

A considerable effort has been made in investigating the factors that determine the evolvability of GasNets (Smith et al. [2002d], Smith et al. [2002c], Smith et al. [2002a], Philippides et al. [2002], Smith et al. [2002e], Smith et al. [2003b], Philippides et al. [2005], Vargas et al. [2008], Husbands et al. [2010]) particularly by Tom Smith and Andrew Philippides in collaboration with Phil Husbands, making use of the concept of neutral networks (see section 2.3.1.4).

Adrian Thompson was successful in extending evolutionary techniques to the evolution of a Field Programmable Gate Array (FPGA), representing one of the first examples of the application of evolutionary techniques to computer hardware

(Harvey and Thompson [1996], Thompson [1996], Thompson [1998]). Ezequiel Di Paulo was responsible for carrying out work on the evolution of spiking neuron models for primitive robot behaviours (Di Paolo [2002], Di Paolo [2003b]), and conducting reserach of a theoretical nature related to homeostasis, embodiment and autopoiesis (Di Paolo [2003a], Izquierdo-Torres and Di Paolo [2005], Di Paolo [2006], Iizuka and Di Paolo [2007], Di Paolo [2008], Di Paolo and Iizuka [2008], Egbert and Di Paolo [2009], Di Paolo [2010]). Theoretical work on the inter-relationship between learning and evolution has also been a subject of research (Harvey [1993a] Mayley [1996], Quinn et al. [2003], Harvey [1996]), representing a resurgence of interest in the Baldwin effect (see section 2.3.1.4).

Homeostasis (an important concept in Ashby's work in the early cybernetics movement) has also been a focus of research interest by other members at the Sussex (Moioli et al. [2008a], Moioli et al. [2009]). Although some work has been done on the evolution of cooperative behaviour in robots (Quinn et al.,Quinn et al. [2003]), the work is perhaps less extensive than that carried out at some other centres such as at CNR and EPFL. In addition to bipedal (McHale and Husbands [2004c], Harvey et al. [2004a], Vaughan et al. [2004], Harvey et al. [2004b]) and quadrupedal (McHale and Husbands [2004b]) locomotion, insect navigation has also investigated using evolved networks (Dale and Collett [2001]). More recently a range of other models have been explored as the basis for cognitive activity, including; reaction diffusion controllers (Dale and Husbands [2010]), spiking networks (Bush et al. [2010]), coupled oscillators Moioli et al. [2010], and explorations of the potential use of chaotic dynamics (Shim and Husbands [2012]).

2.4.1.2 The National Research Council (CNR), Rome

As mentioned previously, Stefano Nolfi and Domenico Parisi based at the National Research Council made a significant contribution to early work in evolutionary robotics (Parisi et al. [1992], Cecconi et al. [1994], Nolfi et al. [1994a], Nolfi et al. [1994b], Parisi and Nolfi [1994], Nolfi and Parisi [1995a], Nolfi and Parisi [1995c], Nolfi and Parisi [1995d], Miglino et al. [1996], Nolfi and Parisi [1997a], Nolfi and Parisi [1997b]), although it is probably fair to say that both Sussex and EPFL have collectively made greater advances in the addressing the practical applica-

tions this research. This is perhaps understandable given that both Stefano Nolfi and Domenico Parisi are members of the CNR's Institute of Psychology. That is not intended to understate their contributions, which have been significant. Work carried out at he institute has addressed; the interaction of learning and evolution, the emergence of language (Parisi [2006], Tuci et al. [2011]), as well as cooperative or group activity in evolved robots (Baldassarre et al. [2003a], Baldassarre et al. [2003b], Dorigo et al. [2004], Baldassarre et al. [2007], Sperati et al. [2011], Trianni and Nolfi [2011]).

The CNR has produced a number of novel models, where adaptation takes place in a population of neural networks. Daniele Denaro produced a novel model of adaptation in neural networks, through incorporating features of both Mendellian and Lamarckian inheritence (Denaro and Parisi [1997]). Lamarkian inheritance, whereby the life-time experience of parents is transmitted to their progeny, is supported by the model a a *culture*. Angelo Cangelosi proposed a model where a simple *language* is evolved and used as the basis of communication between different population members (Cangelosi and Parisi [1998]). The issue of modularity (see section 2.3.1.4) in neural networks has also been addressed (Calabretta et al. [2000], Ferdinando et al. [2000]).

A number of more theoretical issues have also been the subject of study. This includes; ideas relating to the importance of differing time-sacles on the emergence of embodied cognition (Nolfi et al. [2002]), Piagets theory of cognitive development (Mueller and Grobman [2003]), action-based theory of categorization from a connectionist perspective (Borghi et al. [2005]), a comparison of designed and evolved control solutions (Baldassarre and Nolfi [2007]), a perspective on connectionism in artifcial life (Schlesinger and Parisi [2007]), and the evolution of prediction in an embodied agents (Gigliotta et al. [2011]).

2.4.1.3 The Federal Polytechnic of Lausanne (EPFL)

EPFL (under Francesco Mondada and Dario Floreano) and Sussex, have probably contributed the most in early research on the application of evolutionary robotics techniques to real robots. However EPFL was slightly delayed in its entry to this field compared to Sussex and CNR. The first evolutionary robotics papers pro-

duced by EPFL were in the mid 1990's (Floreano and Mondada [1994], Mondada and Floreano [1995]) rather than 1992 as in the case of Sussex and CNR. However work in this area was increased quite significantly towards the end of the 1990's (Nolfi and Floreano [1998], Floreano and Mondada [1998], Urzelai et al. [1998], Floreano and Urzelai [2000a], Urzelai. and J. [2000]). Of particular interest (from the perspective of this thesis) was the development of new forms of neural network capable of online or dynamic adaptation, work which was conducted primarily by Joseba Urzelai. Whilst neuromodulatory effects support plasticity in the GasNet model (originated by Phil Husbands at Sussex), Urzelai was to propose the evolution of networks capable of adapting dynamically via Hebbian learning (Urzelai and Floreano [2001], Floreano and Urzelai [2001a], Floreano and Urzelai [2001b]). With the departure of Joseba Urzelai in 2000 this work has not been investigated nor extended to the extent that the work on the GasNet has at Sussex. Despite this, neural network plasticity, during the life-time of the robot, was to remained a theme of reserach at the laboratory (Floreano and Urzelai [2001b]). Particularly impressive is work conducted at EPFL on a range of real robots; including small wheeled robots, and a small aerial airship (Floreano et al. [2005], Zufferey et al. [2006]) using controllers evolved from networks of spiking neurons. Like Sussex, EPFL has also done some work on collective robotics (Hauert et al. [2008], Mitri et al. [2011]), but again, not to the same extent as that conducted by CNR.

Dario Floreano has done much to publicize the field of evolutionary robotics, writing a number of reviews and general papers, describing work done in evolutionary robotics (Floreano and Urzelai [2000b], Nolfi and Floreano [2002], Floreano et al. [2004a], Floreano et al. [2004b], Floreano et al. [2008], Floreano and Nolfi [2008], Floreano and Keller [2010]).

2.4.2 America

2.4.2.1 Randall D. Beer et al

Randall D. Beer (formerly at Case Western Reserve University, currently at Indiana University), is arguably the leading evolutionary robotics researcher in the USA. His focus on the behavioural aspects of artificial agents is indicated by his proposal of the notion of computational neuroethology (Beer [1990]), indepen-

dently of D. Cliff at Sussex (see section 2.3.4.2), and his interest in the simulation of adaptive behaviour (Chiel and Beer [1991]). He should rightly be regarded as one of the fathers of evolutionary robotics (see section 2.3.5.3), due to the publication of a seminal paper in Adaptive Behavior (Beer and Gallagher [1992b]) in which he described the simulation of primitive behaviours including chemotaxis, locomotion using these techniques. His use of dynamical neural networks was a highly influential methodological development that became common throughout evolutionary robotics.

Randall D. Beer has been a strong advocate of the dynamic systems perspective in considering both the analysis and synthesis of autonomous agents (Beer [1992], Beer [1997], Beer [1998], Beer [2000],Beer [b]). He is also supportive of the ecological-darwinian and autopoietic philosophies underlying the notion of embedded cognition (Chiel and Beer [1997]), Beer [2004]), and has produced papers that integrate these two perspectives (Williams et al. [2008], Beer [2008], Beer [a]). Randall D. Beer's emphasis on dynamics, however, delineates his work somewhat from that of Phil Husbands, Inman Harvey, and Ezequiel Di Paolo at Sussex, whose emphasis is more on Ashby inspired cybernetics and autopoiesis (see section 2.4.1.1).

Randall D. Beer, has been an advocate of some of the simplest models dynamic neural networks (Beer and Gallagher [1992b]); including continuous-time recurrent neural networks (Beer [1995a]) and centre-crossing recurrent neural networks (Mathayomchan and Beer [2002b]). He, together with his co-researchers have applied them to a wide variety of problems. This includes; reactive and sequential learning behaviour (Yamauchi and Beer [1994b], Yamauchi and Beer [1994c]), minimal cognitive behaviour (Beer [1996]), the evolution of a controller for a hexapod robot (Gallagher et al. [1996a]), the evolution of walking (Seys and Beer [2004]), and chemotaxis in a model of a nematode work (Beer [2011]).

His work has also addressed a number of more general issues in evolutionary robotics; such as the role of symmetry generation and genotype size in influencing evolvability (Seys and Beer [2006]), the parameter space structure of continuous-time recurrent neural networks (Beer [2006]), issues of sparse connectivity and in neural network architectures (Psujek et al. [2006]), exaption (Seys and Beer [2007]) and developmental models of genotype to phenotype mapping (Psujek

and Beer [2008]).

Randall D. Beer started with the simulation on minimal cognitive behaviour with a focus on locomotion. Having found solutions to these problems, more recently he has been investigating issues that might help support more complex behaviour, through the synthesis of neural models that address issues of category learning (Williams et al. [2008]) and associative memory (Phattanasri et al. [2007b], Izquierdo et al. [2008]).

These developments parallel the work conducted in this thesis. This thesis starts with an initial focus on locomotion with a comparative study of the dynamic neural network approaches used by Randall D. Beer, Sussex and EPFL. Following this, the focus shifts to models of neural networks supportive of reinforcement learning, that are tractable to incorporation within an evolutionary robotics methodology. This is largely indicative of the development of the field of evolutionary robotics in general; initially focussing on locomotion, and then seeking to extend this to circuit forms that might be supportive of more complex behaviour.

2.4.2.2 Jordan B. Pollack et al

Jordan B. Pollack (initially at The Ohio State University, now at Brandeis University) is of the same generation as Randall D. Beer. His work was originally connectionist with a distinct flavour of classical AI (Pollack [1988]). However in the 1990's we start to see work related to the simulation of evolutionary processes (Watson and Pollack [1992],Angeline and Pollack [1993],Angeline and Pollack [1994]), before addressing the evolution of neural networks (Angeline et al. [1994]). His group conducted one of the earliest works on the evolution of communication (Saunders and Pollack [1996]) prior to that conducted by Cangelosi at CNR (Cangelosi and Parisi [1998]). It is perhaps only towards the end of the 1990's however when he began to focus more on evolutionary robotics (Ficici et al. [1999], Watson and Pollack [1999]). As such, it is probably fair to say that his role has been less significant in the early development of evolutionary robotics than that of Randall D. Beer. However, where he has been influential is in his role as a mentor to a second generation of researchers, including; Richard Watson

(Watson et al. [2002]), Hod Lispon (Lipson and Pollack [2000]) and Greg Hornby (Hornby and Pollack [2001b], Hornby and Pollack [2001a], Hornby and Pollack [2001c]) He continues to do innovative work in collaboration with more junior researchers (Harrington and Pollack [2010], Harrington et al. [2012]).

2.4.2.3 Hod Lipson et al

Hid Lipson, worked as post-doctoral researcher for three years at Brandeis University supervised by Jordan B. Pollack. He then moved to Cornell University, before becoming Associate Director, pf Mechanical and Aerospace Engineering in 2010. According to his personal website;

> My relatively broad spectrum of research projects focus on what I consider to be two *grand challenges* of engineering: (a) Can we design machines that can design other machines, and (b) Can we make machines that can make other machines.

This agenda is evident in the novel robots that he has developed, starting with three generations of simple robots (Pollack et al. [2001]), termed respectively Legobots, GOLEM, and Tinkerbots. This early work was followed by the evolution of a control system for a hovering ornithopter (Regan et al. [2002]) and the evolution of gaits in a hexapedal robot (Bongard [2004]). This work has been reviewed in a paper entitled 'Evolutionary Robotics for Legged Machines : From Simulation to Physical Reality' (Lipson et al. [2006]). Given the grand challenges that he has outlined, it is perhaps not surprising that a significant proportion of his research efforts have focussed on issues of modularity (Lipson et al. [2001] Lipson et al. [2002] Variano and Lipson [2004]) and hierarchy (Lipson [2007]), together with practical work on self-replicating (Mytilinaios et al. [2004] Zykov et al. [2007]) and modular robots (White et al. [2005]). He continues to address novel physical robotics projects with work on the simulation of amorphous robots (Hiller and Lipson [2010]) and the evolution of arbitrary shapes (Clune and Lipson [2011]).

2.4.2.4 Josh Bongard et al

Josh Bongard has had a broad-based education, having passed through The University of Sussex, the University of Zurich, and Cornell University, before arriving at the University of Vermont. As such he has had the opportunity to work at two major centres of evolutionary robotics research (Sussex and Cornell), together with the University of Zurich which is a major centre for autonomous robotics research.

Whilst his early work was significant in its attempts to evolve both morphology and neural control systems (Bongard [2003]), his later work has deviated from the biological imperative emphasized at Sussex. He has adopted a hybrid approach where self-modelling is incorporated as a design feature of his robot control systems (Bongard et al. [2006]).

The fundamentalist form of the evolutionary robotics seeks to discover the processes by which evolution has arrived at intelligence as an adaptation. This not only informs us about biological processes, but it also allows us to explores the dynamics of a process which has open-ended complexity as one of its properties. By introducing a pre-conceived notion of self-modelling, without discovering the process by which such self-modelling might emerge in biological systems, risks abandoning the potential of evolutionary robotics. Although it could be argued that this hybrid approach allows us to achieve greater performance than that which can be achieved through purely evolutionary methods. This may well be justified from an engineering perspective, where a specific problem needs to be resolved in the shortest possible time-scale. It also may be a pre-cursor to hybrid systems that incorporate evolved and designed components. As such, it is worthy of ongoing consideration. Josh Bongard's work represents one of the few examples of a hybrid approach, to problems of non-trivial locomotion and minimal cognitive systems.

Bongard has explored the use of Kenneth Stanley's work on NEAT and CPPNs (Auerbach and Bongard [2010], (Auerbach and Bongard [2011]). He has also conducted more theoretical work in efforts to improve the efficiency of the evolutionary search process; crossover functions for genetic programming (Bongard [2010]), methods to avoid premature convergence (Bongard and Hornby [2010]),

adaptive fitness to improve early termination (Bongard [2011]) and structural modularity (Bongard [2011]).

2.4.2.5 Kenneth Stanley et al

Kenneth Stanley started his work at the University of Texas at Austin, under the supervision of Risto Miikkulainen, before moving to the University of Central Florida. He is arguably one of the leading researchers in the evolution of neural networks in support of minimally cognitive behaviour. His work tends to focus more on the kind of simple simulated environments typically used artificial life research, than on either physically simulation or real robotics. Areas of application include the video entertainment industry, where evolved networks are used to controle non-playable characters in video or computer games (Hastings et al. [2009],Schrum and Miikkulainen [2009]).

Stanley's initially proposed an approach to evolving neural networks, that he labelled NEAT, standing for the NeuroEvolution of Augmenting Topologies (Stanley [2002b], Stanley [2002a]). His work has been highly influential, perhaps largely due to his practice of releasing the source code to his simulations, and placing them in the public domain. This has resulted in a proliferation of his techniques, resulting in the emergence of a number of variants on his original work (e.g. pruning James and Tucker [2004]).

Of particular interest in his approach is the use of; a recombination operator that identifies suitable crossover points in a fashion that preserves some functional characteristics, an approach to speciation that avoids premature convergence, and an incremental approach to network complexity. The NEAT model evolves both network weights and topology (as do the GasNet model developed at Sussex and the Hebbian neural networks developed at EPFL). In addition to node weight and connection information, the genotype also includes a historical markers for each gene. These markers are used as the basis for determining homologous crossover points during recombination. Such a scheme tends to support functional modularity. The same historical markings are used as the basis for a measure of similarity between genotypes. This measure of similarity is then used to associate each geneotype with a nominal species. The evolutionary algorithm

then uses *explicit fitness sharing* (Goldberg and Richardson [1987]), where the fitness of each individual is scaled relative to its performance *within* its species. As a consequence, a genotype of low global fitness (i.e. *absolute* fitness compared with *all* population members), but with high *relative* fitness, compared to other members of its own species, still has the opportunity to reproduce. This helps to maintain phenotype diversity and avoid premature convergence during the evolutionary process. Similar techniques are commonly in multi-objective or multi-modal function approximation (Petrowski [1996], Miller and Shaw [1996], Sareni and Krahenbuhl [1998], Feng and Li [2006], Hsieh et al. [2008]). Instead of seeding the population with networks that have random topologies, the initial population of genotypes is uniform and minimalistic, networks are gradually *augmented* through *add node* and *add connection* mutation operators. The problem domain initially addressed by Stanley was that of pole-balancing. The evolved networks were feed-forward networks. This circuit form is addressed at function apporximation, and through the evolutionary algorithm, function optimization. In contrast, Randall D. Beer, Sussex and EPFL use dynamic neural network variants for dynamic control and sensor processing. These forms are more appropriate for the simulation of sensor-motor control systems in simulated robots.

Although initially an advocate of direct encoding (as initially employed in the NEAT model), Stanley went on to conduct an extensive review of more complex mappings analogous to developmental processes (Stanley and Miikkulainen [2003]). Subsequent to this, Stanley developed what he termed Compositional Pattern Producing Networks (CPPNs) for use in the evolution of complex two-dimensional patterns (Stanley [2007]).

Based on this work he subsequently proposed a model termed HyperNeat (Stanley et al. [2009]) which employs an abstract developmental model in the process of mapping a neural network genotype to its corresponding phenotype (i.e. an *ontogenetic function*). Under this model, an extension of the CPNNs are used not for generating two-dimensional patterns, but for creating connectivity patterns in spatially distributed nodes. The CPNNs themselves are subject to evolution under an extension of the NEAT methodology.

In the generation of a two-dimensional patterns the original CPNNs take a single point as their input. The key innovation in HyperNeat is to take *two*

points, that correspond to the coordinates of two (neuron) nodes. The output of the CPNN then represents a measure of connectivity. If this value is below a certain threshold then the two nodes are *not* connected. If above this threshold, the value is scaled between one and some maximum value, such that it defines a connectivity *weight*. This approach can be generalized to take into account an arbitrary coordinate space for the underlying nodes (for example node distributions in grids, layers, or distributed in concentric circles). The use of this technique has been successfully applied to problems of visual discrimination and food gathering in a simple simulated environment (Stanley et al. [2009]).

Despite the high quality of his work, at least for the purposes of this thesis, his early work is less applicable than that of work conducted by Randall D. Beer, ar at Sussex and EPFL. The reason for this, is that the network models employed by him, are not of the dynamic variety (deemed to be more suitable for dynamic control of real or simulated robots). As a consequence these techniques had not be used to address problems of locomotion. However, this has changed in the last few years with researchers other than Stanley having applied the HyperNEAT methodology to the evolution of gaits (Clune et al. [2009a],Clune et al. [2009b],Yosinski et al. [2011]).

In the course of work carried out for this thesis, a generative model was initially investigated as a possible solution to generating networks that exhibited different functional characteristics. The conclusion of this work was that the search space of possible topologies is extremely large, and that an exploration of this space might not be particularly efficient. Rather a decision was made to first try and conceive of circuit types that might have complementary functional attributes. Theoretically, this would allow us to explore the space of function interaction between these two (or more) complementary types, through some evolutionary process. We knew from the early work that GasNets (and other networks were suitable for locomotion), we ultimately sought to discover a simple circuit type that might emulate characteristics of reinforcement learning, with a formulation that might be tractable to evolutionary search.

An ideal evolutionary algorithm with matched neural network model would be at its most efficient, if we were to explore the possible space of all behaviours directly. Since this is not practical, our preferred solution is to seek to explore the

space of complementary circuits, whose intrinsic characteristics were adaptive to environmental features in support of locomotion (e.g. GasNets) and rewards (our new model). This we believe is likely to be computationally far more efficient in the pursuit of complex behaviours, than an evolutionary exploration of topological space (comprising both connection weight and connectivity) as addressed by the HyperNEAT model.

We speculate that Stanley has also recognized the inefficiency of this approach, as more recent work has sought to modify the fitness function, so that behavioural *novelty* is the target of evolution (Lehman and Stanley [2011b]). This presents problems for a number of reasons; firstly, it does not change the fact that an evolutionary exploration of a topological space might be inefficient, secondly, it is then necessary to create a secondary stage where we determine which of these novel behaviours is of value, and thirdly that it abandons the biological paradigm. That is not to say that is not possible to create a biologically plausible model that is supportive of behavioural diversity. This might be achieved through some form of negative assortative mating (see section 2.3.1.4), where mate attraction is some principled function of behavioural complementarity. Here we digress away from the central point however, that evolutionary of exploration of topological space may be intrinsically inefficient in seeking discover to behavioural novelty, and thus support the evolution behavioural complexity (for a recent review of techniques to support speciation in behavioral space see Trujillo et al. [2011]).

2.5 The Central Challenge

The central challenge facing evolutionary robotics researchers is to seek to expand the boundary of behaviours that we are able to support through neuroevolution. Most evolutionary robotics experiments typically investigate only one particular neuron model and circuit variety. Given the diversity of neurons and wiring patterns found in the brain, it is not unreasonable to suppose that support for complex behaviour might require some integration of functionally specialized brain regions. In seeking to synthesize comparable systems through evolutionary robotics, there are two key issues that need to be addressed. Firstly, at the micro-level, we need to identify (and if necessary create) neuron and circuit models that

support complementary functionality. Secondly, at the macro-level, we need to create hypotheses that provide us with a framework for considering how such circuits are organized and integrated. The second of these is arguably the harder of the two. We already have a wealth of neural network models which we can draw upon for inspiration regarding networks with complementary functionality. We do not however, have established models of the brain as a single system. Such a model can at best only be a hypothesis, since despite significant advances in theoretical neuroscience, the gap between biological reality and holistic functional models remains extremely wide.

Whilst much of the work described in this thesis addresses the micro-level (i.e. models of neural networks), we also attempt to gradually build a conceptual model of the brain as a single system. This is seen in its most complete form in chapter 12. It represents a 'best-fit' model based on the conclusions ascertained from the areas of study encompassed by this thesis. Such a model has the potential to help us to understand how to approach the co-evolutiuon of complementary neural circuits. We use this model to explain why the two circuit varieties (Gasnets and reward adaptive circuits) that we chose to investigate have intrinsically complementary features. This is something to which we shall return in the penultimate chapter of this thesis.

Our goal is to generate agents capable of exhibiting complex behaviour. We term such agents **adaptive replicators**. In pursuit of this goal we seek to create a synthesis of ideas from three distinct areas; the evolutionary robotics methodology, neuron models and neural networks, and the functional integration of neural sub-systems (illustrated schematically in figure 2.7). At a meta level, we might describe these areas as relating to;

1. **Process:** how we generate adaptive replicators.
2. **Implementation:** how we implement artificial neural systems that confer adaptive behaviour on these replicators.
3. **Organization:** how specialized neural sub-systems are functionally integrated.

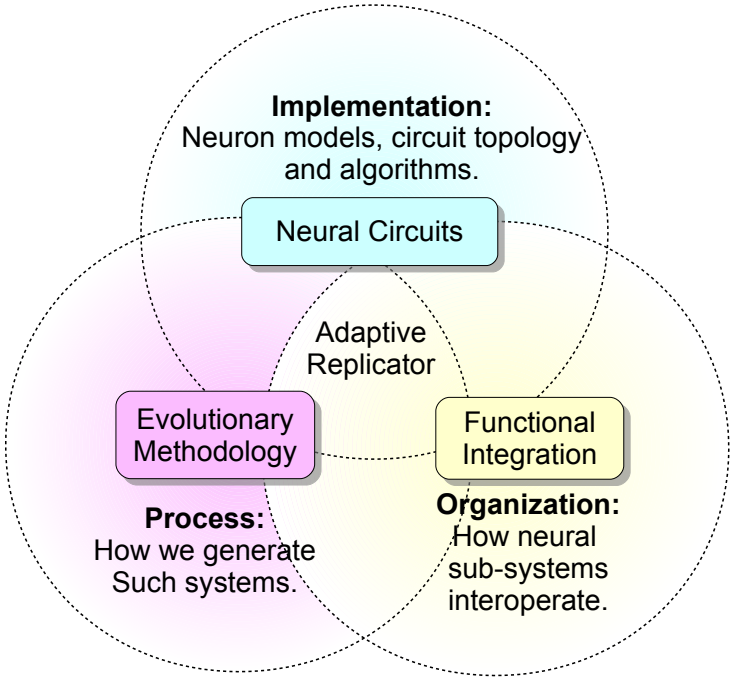


Figure 2.7: Complementary Perspectives

In the following sections we outline some general ideas (that act a high-level context), intended to frame the remainder of this thesis.

2.5.1 Process and Evolutionary Robotics

In seeking to address how greater behavioural complexity might be supported, we need to understand the essential nature of the organism and how it is shaped by its environment. The effect of evolution on replicators, is as a filter, passing through those which act in a manner to promote self-survival and the survival of their progeny. Ultimately all behaviour, is a derivative of the fulfilment of these implied organism goals. In a recursive fashion, neural circuits that support such behaviour are selected for, on the basis that they support the fulfilment of these implied goals. Note that we use to term implied goals here, to distance ourselves from any anthropocentric contamination. There is no intention of implying that organisms act in any conscious sense as if survival and reproduction were their goals, only that evolution will tend to favour those organisms that act *as if* that

were the case.

2.5.1.1 Adaptation as a Unifying Principle

One of nature's key solutions to this problems of competitive replication is the endowment of the organism with properties of *adaptation*. We should be careful to identify the two separate senses in which we the term *adaptation* is used. *Adaptive behaviour* is that which the observer describes when they see an agent responding appropriately to changes in external and internal state. It is a description that corresponds to the perceived actions of the agent from the perspective of a distal observer. *Adaptive processes* in contrast, are those processes that support change in the organism, in support of its implied goals. It is this latter sense in which the term adaptation is of particular interest to us.

There is a very simple way in which we can view adaptive processes; that of a spectrum of processes that result in change in the organism in response to external influences. At one extreme we might describe a process as *generative*, where external influences have a limited impact on the changes that take place in an organism, other than at some point of inception. At the other extreme we might describe processes as *receptive* where the organism undergoes continual change in response to external influences.

2.5.1.2 Adaptation and Evolutionary Robotics

Although this may not necessarily be the view of all evolutionary robotics researchers, it is certainly our view that the evolutionary robotics methodology encompasses this entire spectrum of adaptive processes. Evolution acts on the organism in its entirety, it is in this sense that we use the term evolutionary when we refer to evolutionary robotics. It is through the composition of a number of interacting and coupled adaptive processes that the fitness of an organism is determined. We focus on adaptation as spectrum of processes so as to provide us with a coherent framework from which we are able to consider how disparate processes interact in fulfilment of the same essential goal; that of survival and reproduction. Within this framework, we are able to identify three key classes of adaptive processes; genetic, ontogenetic and continuous adaptation (or learning).

These three classes are depicted in figure 2.8. The classification of any given adaptive process into one class or another is aided by the identification of the principal dimensions in which they differ (as shown in figure 2.9).

One of the key characteristics of the evolutionary robotics methodology is that circuits are evolved and not designed. A neural network circuit is the somatic expression of the germ line cells that incorporate the genetic code. *Direct encoding* ignores this for the sake of methodological expediency. In *direct encoding* the chromosome contains explicit parameters of neurons and their interconnections. Increasingly sophisticated approaches rely on *indirect encoding*, implying a two-stage process. In this case the chromosome codes for parameters of a generative process. It is this generative process that then determines the parameters of and connections between neurons. Indirect encoding incorporates processes that are analogous to ontogenesis in biological systems.

The complexity of this mapping (from chromosome to neural circuit) removes the human agent from any role in explicit design of these circuits. The human role is to ensure that our selected methods of mutation and recombination (that change the genetic code), and the ontogenetic function that maps this code to the neural circuit, has the capacity or theoretical capability of discovering our target design through evolutionary exploration. This involves a convoluted process for a 'robogenetic engineer'. Typically, it is necessary to envisage a target model, that represents a class of systems that might be the result of an evolutionary process. In a traditional engineering approach, we would seek to create a mechanistic decomposition of such a target model, and define the constituent components explicitly. For a robogenetic approach, the decomposition process involves the identification of a simpler class of neurons or networks, that have the capacity, through evolutionary search, to support the functionality that is inherent in our target system.

This approach is reflected in chapters 7 and 8. Chapter 7 develops a hypothetical target model for evolutionary search, that might support reward adaptation. We are then forced however, to identify a far simpler circuit formulation that is capable of acting as a substrate upon which evolutionary search will operate (see chapter 8).

Figure 2.8 represents a bauplan for adaptive replicators. All experiments in

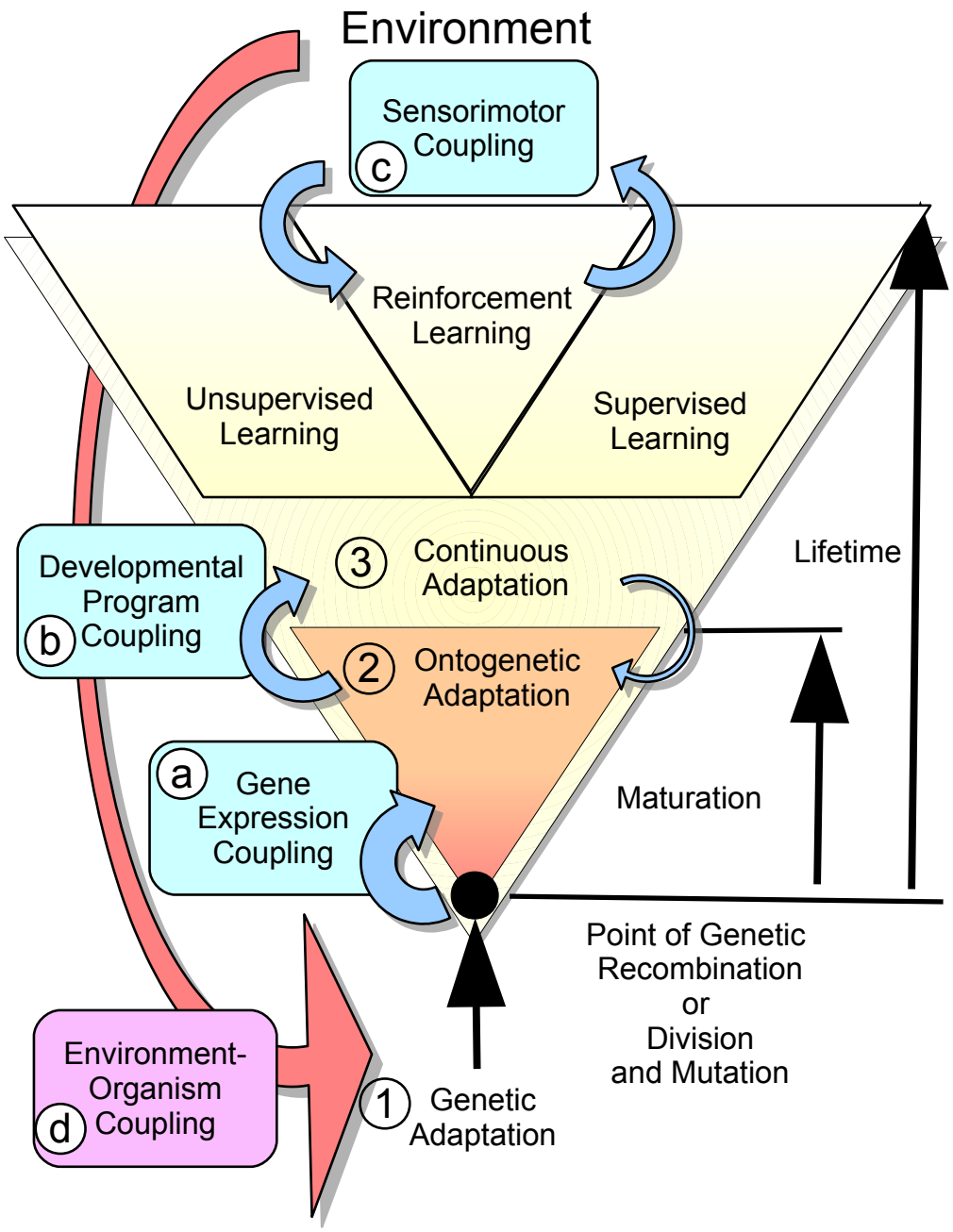


Figure 2.8: A bauplan for adaptive replicators

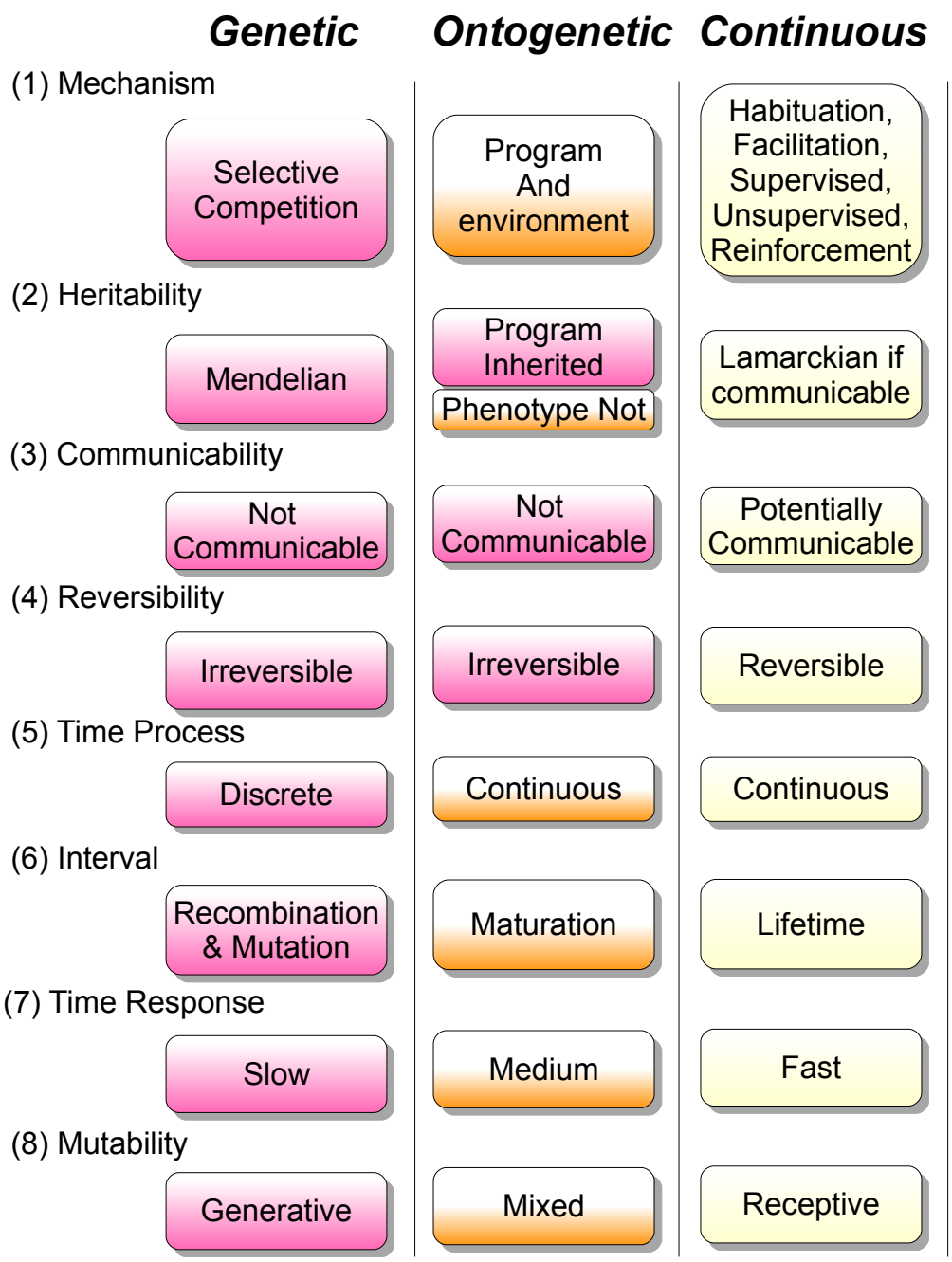


Figure 2.9: The dimensions of adaptation in Evolutionary Robotics

evolutionary robotics incorporate at least one adaptive process. Methods that employ direct encoding (see section 2.2.2) omit ontogentic adaptation, whereas those that employ indirect encoding make use of it (although typically most approaches are entirely generative without being influenced by environmental factors). The more sophisticated the implementation of an adaptive replicator, the greater the number of elements that are incorporated. This bauplan allows us to compare different models of neuroevolution and the circuits that they employ. What it is intended to do however, is also provide us with a macro-level perspective of the processes that interact in the generation of a phenotype capable of exhibiting adaptive behaviour.

2.5.2 Organization and Functional Integration

Evolutionary robotics provides us with a process that allows us to generate adaptive replicators. As a method however it is entirely ambivalent, over the particular results generated, in any respect other than the fitness that a particular neural system might confer to its agent host. As mentioned previously, one of our primary concerns is the efficiency with which the solution space of behaviours might be explored. Without the imposition of some high-level organization that might support the coevolution of functionally complementary circuits, evolutionary exploration may not yield promising results in reasonable time scales. We are particularly interested in the functional integration of specialized neural circuits. Without a macro-level perspective that allows us to envisage some degree of functional specialization, the product of our evolutionary search is likely to be little more than a largely (functionally) undifferentiated neural soup.

Here we are faced with a dilemma. One of the attractions of the evolutionary robotics method is that it supports the discovery of unanticipated designs and open-ended complexity. How do we therefore impose some degree of high-level organization, without over-constraining evolutionary search? A second problem relates to biological plausability. Since this remains our reference model, we have no desire to impose constraints that might preclude the discovery of biologically plausible solutions.

In attempting to consider how we might solve such a problem, we draw inspi-

ration from Ross Ashby's 'Design for a Brain' (Ashby [1960]). He attempted to formulate some general principles of control and organization that might at some fundamental level capture the essence of certain processes that support adaptive behaviour. This is entirely the level of abstraction that is of most use to us. Whilst his work is intended to be biologically plausible, it does not seek to work at the low-level of abstraction typical of neuroscience research, for example. Whilst he proposes a single architecture as an iconic system, he allows for variations on this theme, whereby systems of arbitrary complexity may be coupled together. Whilst we use Ashby's model as a reference, we ultimately arrive at our own formulation, that specifically relates to the primary goal of this thesis. In particular, we are interested in how circuits that are of use in supporting locomotion, may be complemented by those that are subject to reward adaptation. Our general model acts as a framework for explaining how these circuit varieties may be functionally integrated (see chapter 12).

The remainder of this section reviews Ashby's work, and explains why many of the limitations of his model, are avoided through the adoption of the evolutionary robotics methodology. We shall refer to Ashby's ideas intermittently throughout this thesis (briefly in chapter 5 and again in chapter 12). Sections 5.5.1 and 5.5.2 make use of some of Ashby's ideas in modified form, where we seek to identify a mapping from the architecture of a hand-coded solution to a ball collection task to brain anatomy.

2.5.2.1 Ashby's Design for a Brain

Despite being a significant member of the early cybernetics movement, Ashby's perspective differed substantially from those who followed the path of automatic computation. He had little interest in symbols (or representations) and logic as analogies of thought and thinking (see sections 2.3.3 and 2.3.4). Rather, he saw the brain as an organ not dissimilar from that of other organs in the body;

> The biologist must view the brain, not as being the seat of the 'mind', nor as something that 'thinks', but, like every other organ in the body, as a specialised means to survival. (p41, Ashby [1960])

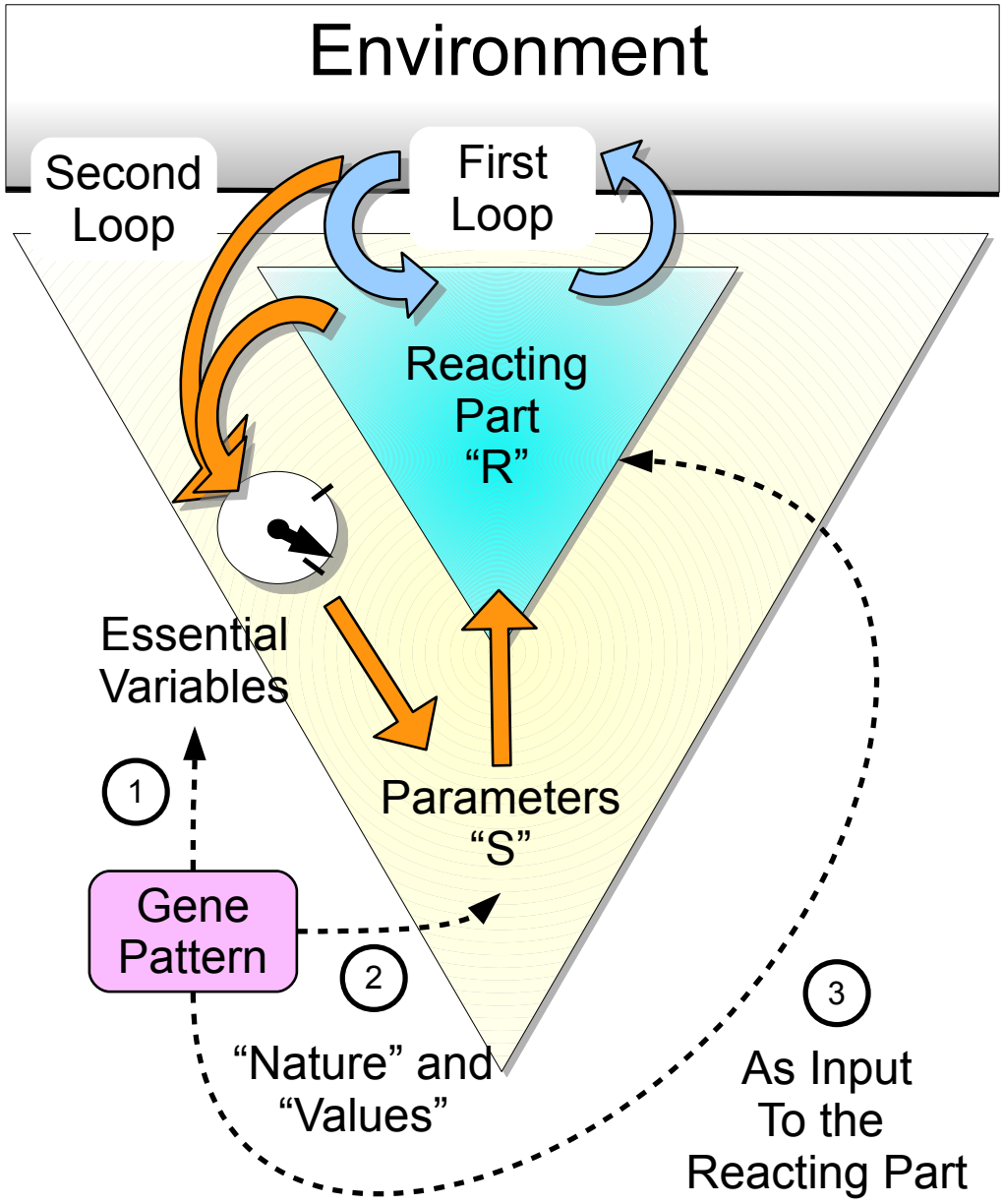


Figure 2.10: Ashby's Notion of Adaptation

He sought rather to arrive at some general principles that might explain the capacity of the organism to exhibit adaptive behaviour. In so doing he formulated a model, comprising a specific architecture, that he termed an *ultrastable* system (p98, Ashby [1960]). This architecture has defined elements, structure and functionality, which ensures that certain **essential variables** (p42, Ashby [1960]) remain within bounds necessary to support the survival of the organism.

Ashby uses the term *environment* to delimit arbitrary divisions within a complex coupled system. He typically uses the term largely to differentiate between that which is inherent in a functional sub-system, and that which is external (although coupled) to it;

> There it is intended to treat one group of neurons in the brain as the environment of another group. These divisions, though arbitrary, are justifiable because we shall always treat the system as a whole, dividing it into parts in this unusual way merely for verbal convenience in description. (p41, Ashby [1960])

Ashby's original definition of an *ultrastable* system is illustrated in figure 2.10. There are two key structural elements which he refers to simply as the *first loop*, and the *second loop*. The *first loop* supports a process, which takes as its input, some feature from the environment, and in turn engages in a suitable response. The system that supports this is referred to as the *reacting part* (R). The second key structural element is that of the *second loop*. The second loop accepts as its input, sensor information from the environment and/or internal information from the organism, and in turn seeks to ensure that certain *essential variables* remain within a range of preferred values. A deviation from these bounds, results in a change of *parameters* (S) that in turn modify the behaviour of the *reacting part* (R). This continues until appropriate *parameters* (S) are selected, such that these *essential variables* are once again within bounds. In Ashby's words;

> For convenience, its definition will be stated formally. Two systems of continuous variables (that we called 'environment' and 'reacting part') interact, so that a primary feedback (through complex sensory and motor channels) exists between them. Another feedback, working intermittently and at a much slower order of speed, goes from the

> environment to certain continuous variables which in their turn affect some step-mechanisms, the effect being that the step-mechanisms change value when and only when these variables pass outside given limits. The step mechanisms affect the reacting part; by acting as parameters to it they determine how it shall react to the environment. (p98, Ashby [1960])

Whilst evolution is not a central focus of his work, it is incorporated to the extent that he perceives three roles for a *gene pattern* (p134, Ashby [1960]);

1. Determining the *essential variables* and their limits.
2. As an input to the *reacting part* (R).
3. In the determination of the *parameters* (S), in both *value* and *form* (i.e. the functional characteristics of the systems that act on the *reacting part*).

2.5.2.2 The Limitations of Ashby's Approach

There are three principal difficulties in adopting Ashby's approach;

1. **Design:** He outlines no principled means by which we might *pre-determine* the capacity or capability of an *ultrastable* system, and thereby ensure that it a given design is appropriate for a given task. In other words, we have no basis for design. We do not know the outcomes that might result from architectural decisions.
2. **Training:** His outline of training an ultrastable system is crude, relying upon the repeated disruption of the system (which he describes as analogous to *punishment*), until the system delivers the required behaviour.
3. **Functional Specialization:** The outlined model fails to identify the role (if any) of specialized functional subsystems, and how they might influence the behaviour exhibited by such a system.

This latter limitation is implicitly acknowledged by Ashby himself, when discussing the challenges of training an *ultratstable* system;

> The concept of 'reward' is more complex. It usually involves the supplying of some substance (e.g. food) or condition (e.g. escape) whose absence would act as 'punishment'. The chief difficulty is that the evidence suggests that the nervous system, especially the mammalian, contains intricate and specialised mechanisms which give the animals properties not to be deduced from basic principles alone. (p110, Ashby [1960]);

Reward adaptive systems are therefore not specifically addressed by his model, since according to Ashby such processes are not easily defined through general models, and therefore beyond the scope of his treatment.

Given that we are able to have arbitrary variants of the ultrastable system, and yet we are unable to identify forms that might support specialized functionality, and we have no efficient means by which we might train such a systems, it is perhaps not surprising that computational approaches we seen as preferential for the purposes of device construction.

2.5.2.3 Evolutionary Robotics as a Solution

The limitations of Ashby's approach are in many ways circumvented by adopting the evolutionary robotics methodology. The capacity or capability of a system is tested during a fitness trial. The use of evolutionary algorithms to explore the space of possible designs avoids the requirement for the need for an analytical means of determining design capability. The role of the human designer is no longer one of directly determining the behaviour of a system, but that of seeking to support the efficient exploration of potential solutions via the process of evolutionary search.

There are no limitations on the nature of the circuits that can be incorporated within an evolutionary robotics solution (other than that they should be tractable to evolutionary search). Circuits that are amenable to continuous adaptation (i.e. learning) can easily be incorporated. This provides us with a means of supporting training in a fashion that is more flexible than that exhibited by the techniques employed by Ashby in his 'training' his homeostat. Finally, circuits of any arbitrary functional specialization can be incorporated into such designs,

without necessarily requiring a human understanding of how such circuits might interact with other sub-systems.

Whilst the computational approach facilitates (and in fact requires) human design input, the results are brittle. Small errors in implementation can result in the non-graceful degradation of results. Ashby's formulation is more attractive to the evolutionary robotics approach, since the inherent hierarchy imposed by the second-loop imposes stability at the macro-level of behaviour, thus avoiding the potentially fatal errors due to minor variations. It thus presents us with a smoother fitness landscape, than that offered through the evolutionary exploration of computational systems alone.

Features of the Ashby approach that are consistent with evolutionary robotics include the genetic determination of certain system features (as illustrated in figure 2.10). Central to both Ashby's notions of adaptation and evolutionary robotics, is the centrality of the *organism* as a model for an artificial control systems and a focus on the capacity of the organism to exhibit appropriate responses to changes in environmental circumstances;

> The book is not a treatise on all cerebral mechanisms but a proposed solution of a specific problem: the origin of the nervous system's unique ability to produce adaptive behaviour. (p v, Ashby [1960])

The evolutionary robotics methodology is inspired by biological systems and the robot itself is seen as some analogue of an organism. However, Ashby's formulation is rather too general for our purposes.

Unlike Ashby, we include *ultrastable genetic adaptation* as a key adaptive process. We would argue that the dual loop characteristic of Ashby's ultrastable systems model is preserved where we treat the gene population as the *reacting part* (R), population culture as the *parameters* (S), and the *essential variables* as those corresponding to energy and food supplies (for example). Such a system is depicted in figure 2.11. Usually, the term *genetic adaptation* is used to refer to changes in allele frequency in a population or a representative individual. We shall continue to use it in that sense. It seems preferable to use the term *ultrastable genetic adaptation* only when we wish to emphasize this dual-loop formulation. The concept of an *ultrastable genetic adaptation* is of value to

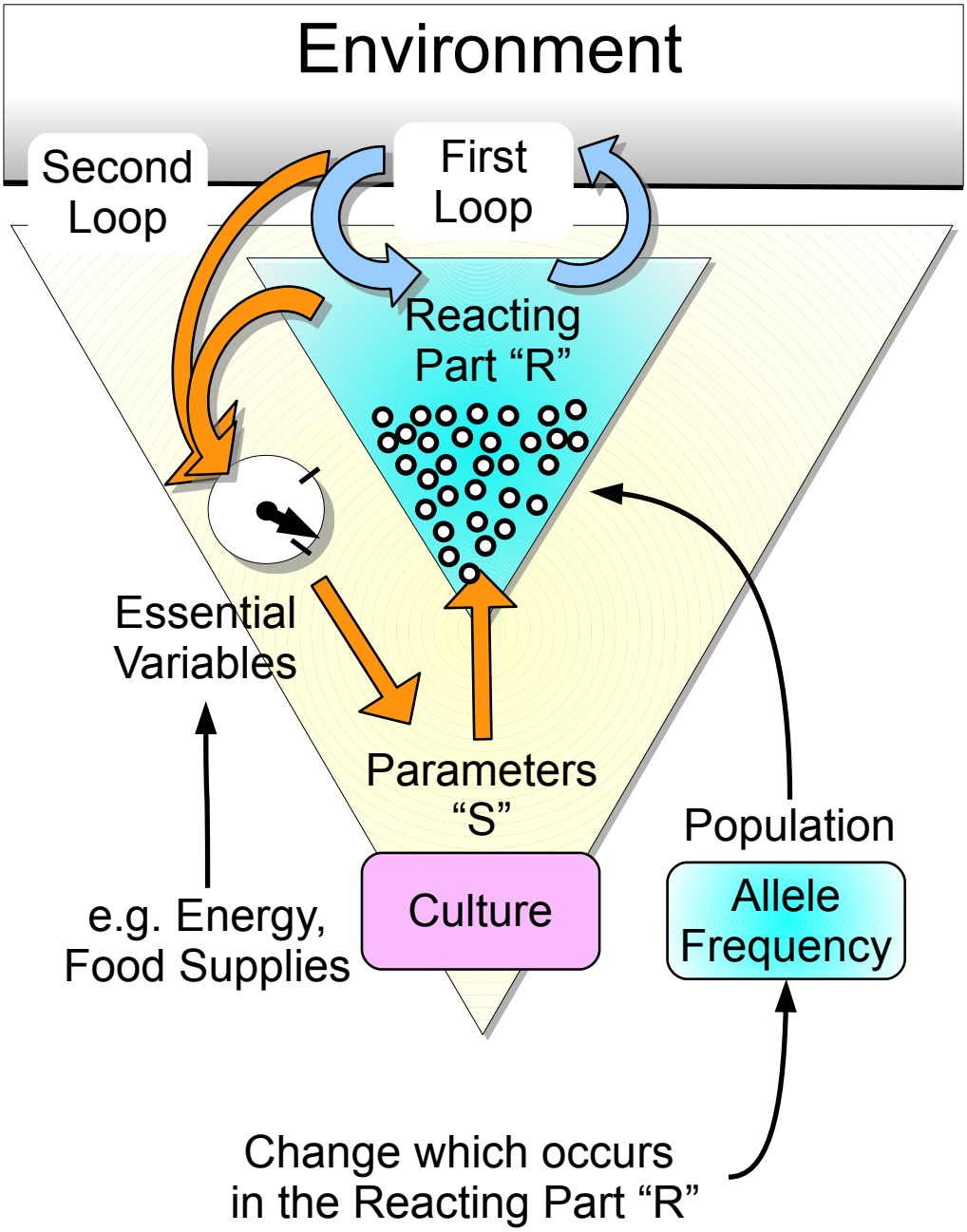


Figure 2.11: An *Ultrastable* System model of Genetic Adaptation

evolutionary robotics since it provides us with a conceptual framework to address issues such as; the emergence of language and communication and the coevolution of memetic systems. However, we shall not specifically deal with these issues in this thesis.

Whilst the system as depicted does not comply with Ashby's definition of an ultrastable system (in that cultural change would not strictly speaking be a continuous process, and that no single *gene pattern* exists as an exogenous input to the system), we would argue that it can be regarded as one if certain conditions are relaxed.

The ontogenetic domain is of particular relevance to evolutionary robotics, since it is through the *ontogenetic function* (see section 2.2.2) that the genotype corresponding to an individual is mapped to the corresponding neural network. Models that employ direct-encoding employ trivial mappings that bear little resemblance to the processes of *gene expression* and *developmental programs*. However formulations that make use of indirect encoding typically correspond to some analogue of these processes.

It is the generative properties of the ontogenetic domain that determine those processes that are created in the domain of continuous adaptation. At least for the purposes of evolutionary robotics, we can identify a number of specialized circuits that comprise examples of adaptive sub-systems, that in conjunction support the organism in the maintenance of essential variables. Examples of such sub-systems include; *unsupervised learning* and the creation of mappings from high-dimensional to low-dimensional spaces and feature clustering; *supervised learning* for function approximation in a system that is rich in error feedback signals; *reinforcement learning* and the determination of beneficial action sequences in a high-dimensional space. The domain of continuous adaptation might also encompasses less primitive forms of learning such as facilitation and habituation, as well as specialized circuits such as Hopfield networks for associative memory.

Our bauplan for an *adaptive replicator* (figure 2.8) represents a solution space of possible architectures, under a modified notion of adaptation as used by Ashby. In our pursuit of the central challenge of evolutionary robotics, are task is clear. It is to determine or create; efficient methods for the exploration of this solution space; models that support complementary functional activity. There is an in-

trinsic order in which these issues can be addressed. We need to first determine what circuits embody the complementary functional characteristics required to support complex adaptive behaviour. The identification of suitable neuron models and circuits that are tractable to evolutionary search remains one of the core areas of research in evolutionary robotics, and it is this issue that represents the primary goal of this thesis. Once such circuits have been identified, future work (beyond the scope of this thesis) will be necessary to discover how best these circuits might be integrated through coevolution and exaption.

2.5.3 Implementation and Neural Networks

A key element in supporting complex behaviours is in identifying (and if necessary creating) functional circuits that exhibit complementary characteristics, whose key parameters can be explored via evolutionary search. Whilst our long-term goal might be to integrate such circuits, our initial goal has to be that of identifying which circuits might be of particular interest. Our approach to answering this question is to consider different problem domains, and then establish which circuit varieties are particularly suited to them. In particular we focus on locomotion and reward adaptation;

- **Locomotion:** As we saw in section 2.4 the application of evolutionary robotics to problems in locomotion is already well developed. Principal amongst the approaches to supporting locomotion are neural network formulations that include; GasNets (Husbands et al. [1998e]), continuous-time recurrent neural networks (Mathayomchan and Beer [2002b]) and Hebbian networks (Urzelai and Floreano [2001]). The starting point for research described in this thesis was therefore to identify which of these models best at supported locomotion in legged robots.

  In these studies we sought to determine which of the features of these various formulations are significant in influencing their performance. This involved an extensive comparative study, using 12 computers with simulation times alone requiring six months. A description of these comparative studies is given in the next chapter (chapter 3).

- **Sequence-Learning via Reward Adaptation:** Having established GasNets to be of particular use in supporting locomotion, we sought to apply them to an alternative task; that of having a robot collect a number of balls (chapter 4). Disappointing results prompted us to reconsider how such tasks might best be solved through circuits tractable to evolution (chapters 5 and 7). The result was an investigation of sequence-learning via reward adaptation (chapters 9 and 10). This work is based on novel neuron model and circuits that incorporate them (chapter 8) inspired by the role of dopamine in the basal ganglia (chapter 6).

Chapter 3

Evolving Locomotion - Comparative Studies

3.1 Introduction

3.1.1 Overview

This chapter describes two comparative studies designed to establish which formulation of dynamic neural networks are best suited to the evolution of locomotion in two physically simulated robots; a biped (with restricted degrees of freedom) and a quadruped (with unrestricted freedom). It compares GasNets developed at Sussex (see section 2.4.1.1), Plastic or Neural Networks developed (PNNs) at EPLF (see section 2.4.1.3), Centre-crossing Continuous Time Recurrent Networks (CCTRNs) developed at Case Western Reserve University (see section 2.4.2.1) together with variations and combinations of these basic types. These different formulations were chosen as the leading examples of evolvable networks in support of locomotion, GasNets originally developed by Phil Husbands (Husbands [1998b], Husbands et al. [1998d]), PNNs developed by Urzelai and Florean (Urzelai and Floreano [2001], Floreano and Urzelai [2001a]), and CCTRNs based on models proposed by Randall D. Beer. (Beer [1996], Mathayomchan and Beer [2002b]). At the time of writing no significant studies had addressed this issue. The simulation time alone required approximately 6 months using up to 12 computers; 8 computers with Intel based CPU's running at 2Ghz and 4 computers

running at 3GHz.

There is an increasing body of work that has successfully applied these techniques to the evolution of ANNs suitable for controlling bipedal and quadrupedal locomotion. This includes earlier work (Reil and Husbands [2002]) which demonstrates that it is possible to evolve a bipedal motor control in a physically simulated agent using a conventional Dynamic Recurrent Neural Network (DRNN) without sensor input. Bongard and Paul have evolved bipedal locomotion in a physically simulated agent through genetic encoding that comprises morphological as well as ANN parameters (Bongard and Paul [2001]). Researchers have evolved bipedal locomotion in a physically simulated robot that incorporates a model of neuromodulation (Ishiguro et al. [2003a]). Billard and Ijspeert have been successful in evolving quadrupedal locomotion in a real robot (Billard and Ijspeert [2000]).

3.1.2 The Dimensions of Comparison

The first comparative study (section 3.2) seeks to compare 14 different varieties of networks in support of bipedal locomotion. The reason for choosing such a large number of varieties was to seek to determine heuristically which circuit features might be significant in determining the evolvability of the networks. The second comparative study sought to focus on the simplest implementations of these circuits in support of quadrupedal locomotion (section 3.3). In doing so we sought to explore the potential benefits or detriments of;

- **Connectivity:** Fully recurrent versus sparse connectivity (in the GasNet model).
- **Neuromodulation:** The GasNet model of neural modulation (GasNet with and without gas neuromodualtion).
- **Hebbian Plasticity:** The use of dynamically variable weights based on Hebbian learning rules.
- **Density:** Neuron density (16 versus 32 neurons the GasNet model).

- **Hybridization:** Hybrid variants (GasNet-PNN, CTRNN-PNN, GasNet-CTRNN).
- **Generality:** The simplest forms of GasNet, CTRNN, PNN networks are tested on both a biped and quadruped model in order to see if morphology had a significant impact on network variant performance.

One significant area where we seek to maintain everything constant, is in the genetic algorithm used. This study is not intended to say anything about the optimal genetic algorithm that might be used with such networks.

3.1.3 Common Methods

For the purposes of comparison the genetic algorithm employed is invariant across all tests. These comparative studies are not intended to address the performance characteristics of differing genetic or evolutionary algorithms, but rather the features of different formulations of neural networks.

3.1.3.1 Genetic Algorithm

The same distributed steady-state genetic algorithm is used on all networks. The population grid has dimensions of 10 by 10, to yield a total of 100 individuals. Competition is tournament based, with a tournament group comprising three individuals. A *principal* is selected at random from the grid. Two other population members are selected by a random walk originating at the principal's grid cell. The length of the random walk is an integer value in the range 1-4.

If the principal is the fittest then the weakest member is replaced by a mutated version of the principal. Otherwise the weakest member of the tournament set is replaced by a recombination of the two fittest individuals (using single point cross-over). This recombined genome is then mutated.

In these experiments the generation index is incremented after the evaluation of 100 individuals (comprising a pseudo-generation). Each neural network type was evaluated for 200 generations. This was carried out ten times with different random seeds for each network type.

Note that the approach to mutation (the frequent modification of parameters by a small amount, and a less frequent mutation by a substantially greater amount) is in line with previous GasNet experiments and that the use of an asynchronous distributed style GA is intended to emulate early GasNet experiments (Husbands et al. [1998e]). One significant difference from that of the early work on GasNets, is that we are employing single-point cross-over as a recombination operator. Early GasNet work relies exclusively upon mutation operators (Husbands et al. [1998e]). Note that in later work on GasNets a comparison of the use of GasNet models with and without recombination operators was made; recombination yields slightly superior results than the sole use of mutation operators (Smith et al. [2002e]). Later GasNet experiments (such as in Husbands et al. [2001b] and Philippides et al. [2002]) would use node addition and deletion operators, which are not employed here. The number of nodes used in a given network type is fixed. One of the reasons for this, is that we want to carry out simulations where we compare the variation node density, in an attempt to determine whether or not this is a significant factor in determining the performance of a GasNet.

3.1.3.2 Genetic Encoding and Mutation

The genetic encoding strategy follows a similar approach for all networks. Network parameters are stored on a node or cell basis. Each gene comprises a list of real valued and integer parameters (comprising 16 parameters per node for a conventional GasNet for example). Connection weights (where relevant) are also stored on a per node basis.

Mutation takes place either after recombination, or after cloning of the principal tournament member (as described earlier). Mutation takes place at 20 percent of the nodes (rounded to 3 in a 16 cell network) selected at random. A single mutation event will result in the mutation of a single real or integer parameter in each of the randomly selected nodes. The magnitude of this mutation corresponds to 4 percent of the real valued parameters range with a probability of 0.2, and 1 percent of the parameters range with a probability of 0.8. In the case of integer parameters we follow a similar strategy of small mutations with a probability

of 0.8 and large mutations with a probability of 0.2. These mutation parameters were chosen in preliminary experiments to avoid premature convergence and maintain a reasonable degree of phenotypic diversity across the different network varieties during evolution.

In addition to this, those networks where connection weights are under evolutionary control (such as in CTRNNs) undergo further mutation. Each randomly selected cell has all of its weights mutated (again by a factor of 4 percent with a 20 percent probability and 1 percent with an 80 percent probability).

Time constant initialization was devised to yield a wide range of of values. An exponent f was randomly selected from the set:

$$f\epsilon[-10, -8, -6, -4, -2, 0, 2, 4, 8, 10]$$

A second random variable $r\epsilon[0.0, 1.0]$ was then used to scale the value such that the time τ constant is calculated from:

$$\tau_i = 1.0 + r_i(10^{f_i}) \tag{3.1}$$

The time constant mutation operator increments or decrements the exponent by 1 with a probability of 0.2, and generates a new value of $r\epsilon[0.0, 1.0]$ with a probability of 0.8.

3.1.4 Characteristic Equations

3.1.4.1 Center-Crossing CTRNNs

The characteristic equation of the conventional CTRNN (Beer [1995b]) is shown below;

$$y_i^{t+1} = y_i^t + \frac{T}{\tau_i}(-y_i^t + \sum_{j=1}^{N} \omega_{ji}\sigma(y_j^t + \theta_j) + I_i) \tag{3.2}$$

$$i = 1, 2, ..., N$$

Where:

y_i^{t+1} is the activation of the i'th node at time $t + 1$.

y_i^t is the activation of the i'th node at time t.

τ_i is the time constant for the i'th node calculated according to equation 3.2.

I_i a sensor input to the i'th node where I is either 1 (in contact with the floor) or 0 (not in contact with the floor).

θ_j a bias term for the j'th node where $\theta \in [-2, 2]$.

T is the time slice (in this case T is set to 1).

ω_{ji} is the weight of the output from the j'th node to the i'th node where $\omega \in [-4.0, 4.0]$.

σ is the logistic activation function.

$$\sigma(z) = \frac{1}{(1 + e^{-z})} \tag{3.3}$$

The network is fully interconnected. Node connection weights and bias are under evolutionary control. This study uses a variant of the conventional CTRNN, referred to as the Center-Crossing CTRNN Mathayomchan and Beer [2002a], where initial biases are calculated such that:

$$\theta_i = \frac{-\sum_{j=1}^{N} \omega_{ji}}{2} \tag{3.4}$$

Mathayomchan and Beer suggest that populations seeded with center-crossing networks may be more likely to yield a wider range of dynamics than a population of random networks.

The incorporation of a single symmetry axis (as used in this study) results in a final form, as described by;

$$y_i^{t+1} = y_i^t + \frac{T}{\tau_i}(-y_i^t + \sum_{j=1}^{N} \omega_{ji}\sigma(y_j^t + \theta_j - z_i^t) + I_i) \tag{3.5}$$

$$i = 1, 2, ..., N$$

Where:

z_i^t is the activation of the corresponding i'th node in the symmetrical subnetwork at time t.

N is the number of nodes in each subnetwork (in this case 8).

3.1.4.2 GasNets

GasNets are an example of a class of ANNs that seek to model aspects of neuromodulation. A key attribute of the GasNet model is that the transfer characteristics of network nodes are modified via the influence of diffused gases (modeled in a 2-dimensional plane). This network model is inspired by the action of Nitric Oxide in biological systems (Husbands et al. [2001a]). Earlier work has shown that GasNets are more *evolvable* than comparable networks that do not incorporate gas modulation, both in simulation and when used in real robots (Smith et al. [2003a]).

In GasNets, node transfer functions can be modulated by local gas concentrations in the vicinity of the node. Nodes can also act as chemical emitters, under either gas or electrical stimulation. GasNet nodes exist in a geometric plane where internode distances determine gas concentrations and (in conjunction with additional genetic parameters) network connectivity. Under typical evolutionary parameters the GasNet connectivity rules result in a sparsely connected network.

$$y_i^{t+1} = \tanh[k_i^t(\sum_{j \in C_i} \omega_{ji} y_j^t + I_i) + b_i] \tag{3.6}$$

Where:

k_i^t is a time-varying transfer function modulator. The value of k varies with gas concentrations at the i'th node, see equation 3.10.

C_i is the set of all nodes that have an input to the i'th node.

I_i a sensor input to the i'th node.

b_i a bias term for the i'th node where $b_i \in [-2, 2]$.

The original GasNet diffusion model (upon which this implementation is based) is controlled by two genetically specified parameters, namely the radius of

influence r and the rate of build up and decay s. Spatially, the gas concentration varies as an inverse exponential of the distance from the emitting node with a spread governed by r, with the concentration set to zero for all distances greater than r (Equation 3.7). The maximum concentration at the emitting node is 1.0 and the concentration builds up and decays from this value linearly as defined by Equations 3.8 and 3.9 at a rate determined by s.

$$C(d,t) = \begin{cases} e^{-2d/r} \times T(t) & d < r \\ 0 & \text{else} \end{cases} \tag{3.7}$$

$$T(t) = \begin{cases} H\left(\frac{t-t_e}{s}\right) & \text{emitting} \\ H\left[H\left(\frac{t_s-t_e}{s}\right) - H\left(\frac{t-t_s}{s}\right)\right] & \text{not emitting} \end{cases} \tag{3.8}$$

$$H(x) = \begin{cases} 0 & x \leq 0 \\ x & 0 < x < 1 \\ 1 & \text{else} \end{cases} \tag{3.9}$$

where C(d,t) is the concentration at a distance d from the emitting node at time t. t_e is the time at which emission was last turned on, t_s is the time at which emission was last turned off, and s (controlling the slope of the function T) is genetically determined for each node. The total concentration at a node is then determined by summing the contributions from all other emitting nodes (nodes are not affected by their own concentration, to avoid runaway positive feedback).

For mathematical convenience, in the basic GasNet there are two 'gases', one whose modulatory effect is to increase the transfer function gain parameter (k_i^t from equation 3.6) and one whose effect is to decrease it. It is genetically determined whether or not any given node will emit one of these two gases (gas 1 and gas 2), and under what circumstances emission will occur (either when the 'electrical' activation of the node exceeds a threshold, or the concentration of a genetically determined gas in the vicinity of the node exceeds a threshold. Note these emission processes provide a coupling between the 'electrical' and 'chemical' mechanisms). The concentration-dependent modulation is described by Equation

3.10, with transfer parameters updated on every time step as the network runs.

$$k_i^t = k_i^0 + \alpha C_1^t - \beta C_2^t \tag{3.10}$$

where k_i^0 is the genetically set default value for k_i, C_1^t and C_2^t are the concentrations of gas 1 and gas 2 respectively at node i at time t, and α and β are constants. Both gas concentrations lie in the range $[0, 1]$. Thus the gas does not alter the electrical activity in the network directly, but rather acts by continuously changing the mapping between input and output for individual nodes, either directly or by stimulating the production of further virtual gas. The concentration dependent modulation can, for instance, change a node's output from being positive to being zero or negative, even though the input remains constant. Any node that is exposed to a non zero gas concentration will be modulated. This set of interacting processes provides the potential for highly plastic systems with rich dynamics.

3.1.4.3 CTRNN/PNN Hybrid

One of the underlying concepts associated with Plastic Neural Networks is that there is value in evolving artificial neural networks that are capable of exhibiting learning through ontogenetic change (Floreano and Mondada [1996]). Let us first start with a description of a basic PNN (Urzelai and Floreano [2000a]). A key characteristic of PNN's is that connection weights vary over time based on Hebbian learning rules given by:

$$\omega_{ji}^t = \omega_{ji}^{t-1} + \eta \Delta\omega_{ji} \tag{3.11}$$

Where η is a learning rate ($0.0 < \eta < 1.0$) and ω_{ji} is the connection weight of the input to node i from node j. The adaptation rule $\Delta\omega_{ji}$ is genetically determined for each node. All inputs to a given node are subject to the same adaptation rule (referred to as node encoding by the authors).

Where x is the activation of node j, which is an input to node i (which has an output activation of y), the adaptation rule is one of:

Plain Hebbian Rule

$$\Delta\omega_{ji} = (1 - \omega_{ji})x_j y_i \tag{3.12}$$

Post-Synaptic Rule

$$\Delta\omega_{ji} = \omega_{ji}(-1 + x_j)y_i + (1 - \omega_{ij})x_j y_i \tag{3.13}$$

Pre-Synaptic Rule

$$\Delta\omega_{ji} = \omega_{ji}x_j(-1 + y_i) + (1 - \omega_{ji})x_j y_i \tag{3.14}$$

Covariance Rule

$$\Delta\omega_{ji} = \begin{cases} (1 - \omega_{ji}) & \text{if } F(x_j, y_i) > 0 \\ (\omega_{ji})F(x_j, y_i) & \text{otherwise} \end{cases} \tag{3.15}$$

Where:

$$F(x_j, y_i) = \tanh(4(1 - |x_i - y_j| - 2) \tag{3.16}$$

All nodes in the PNN are fully interconnected. The rate of learning η can only assume one of four values (0.0, 0.3, 0.6, 0.9). The characteristic equation for the PNN is shown below:

$$y_i^{t+1} = \sigma(\sum_{j=1}^{N} \omega_{ji}^t(y_j^t)) + I_i \quad i = 1, 2, ..., N \tag{3.17}$$

Where:

ω_{ji}^t is the adaptive weight for the j'th input to the i'th node.

σ is the standard logistic activation function.

I_i a sensor input to the i'th node where I is either 1 (in contact with the floor) or 0 (not in contact with the floor).

The CTRNN/PNN Hybrid is a variation on the conventional PNN. The difference being that activation signals are further modified by a node based time constant under evolutionary control (in a similar fashion to conventional CTRNNs). This variation was first introduced by the authors of this paper in an attempt to

create a PNN that exhibited richer frequency dynamics than those of the conventional PNN (McHale and Husbands [2004a]). The range of y_i is $[0, 2]$ for input neurons and $[0, 1]$ for hidden and output neurons (Blynel and Floreano [2002]). It is modified again here such that the network comprises two symmetrical subnetworks, with mutual inhibition;

$$y_i^{t+1} = y_i^t + \frac{T}{\tau_i}(-y_i^t + \sum_{j=1}^{N} \omega_{ji}^t \sigma(y_j^t + \theta_j - z_i^t) + I_i) \tag{3.18}$$

$$i = 1, 2, ..., N$$

Where:

ω_{ji}^t is the adaptive weight for the j'th input to the i'th node.

z_i^t is the activation of the corresponding i'th node in the symmetrical subnetwork at time t.

N is the number of nodes in each subnetwork.

3.2 Experiment 1: Bipedal Locomotion

This section starts by describing the experimental set-up and the genetic algorithm used on all networks. It then goes on to list the characteristic equations of each network variety before finally reporting the results. A total of 14 different ANNs are assessed in this study.

3.2.1 Experimental Setup

Parameters were chosen as outlined by the original authors wherever possible (given the differences in genetic encoding). These original papers should be consulted for further experimental details if more information is required. Variations from their implementations are stated where relevant.

3.2.1.1 Ab Initio

For the purposes of a fair comparison no assumption is made about the coupling of the underlying network nodes. The chosen task is to achieve bipedal locomotion *ab initio*, without assuming that nodes should be configured as coupled oscillators (such as those in Matsuoka [1985]).

3.2.1.2 Physical Model

The computer code for the biped physical model was generated with the aid of a product called Autosim (unfortunately no longer commercially available technical details of the product are described in Sayers [1990]). Joints are simulated as torsional springs. Strictly speaking the motor output is actually a control signal. This signal is mapped to an angular displacement that corresponds to the rest position of a torsional spring. A change in the angular displacement of this rest position will result in a torque applied to the lower limb attached to the joint (as the spring seeks to restore the joint to its new rest position).Clearly this is not in any way biologically realistic. Earlier attempts to evolve bipedal locomotion using a model based on antagonistic muscle groups (McHale [2000]) were unsuccessful. The current approach is based on work outlined in Reil (Riel [1999]).

The kinematic root only has five degrees of freedom (two rotational and three translational). The biped is physically incapable of rotating in its roll axis. This prevents the biped from falling over on its side, although it is still free to fall forwards and backwards. Whilst this is not entirely physically realistic, it is sufficient for the purposes of this comparative study. This still remains a non-trivial problem; feet are modelled as point contact points, resulting in a dynamically unstable model after an initial displacement of the biped. Figure 3.1 shows a diagram of the model's dimensions, weights, joint angular limits and torsional spring constants.

3.2.1.3 Fitness Calculation

Each fitness assessment starts with the biped in a stable standing position. The trial lasts a simulated 20 seconds. The fitness of the individual is taken to be the minimum of the distance travelled by either of the biped's feet, or hips. This

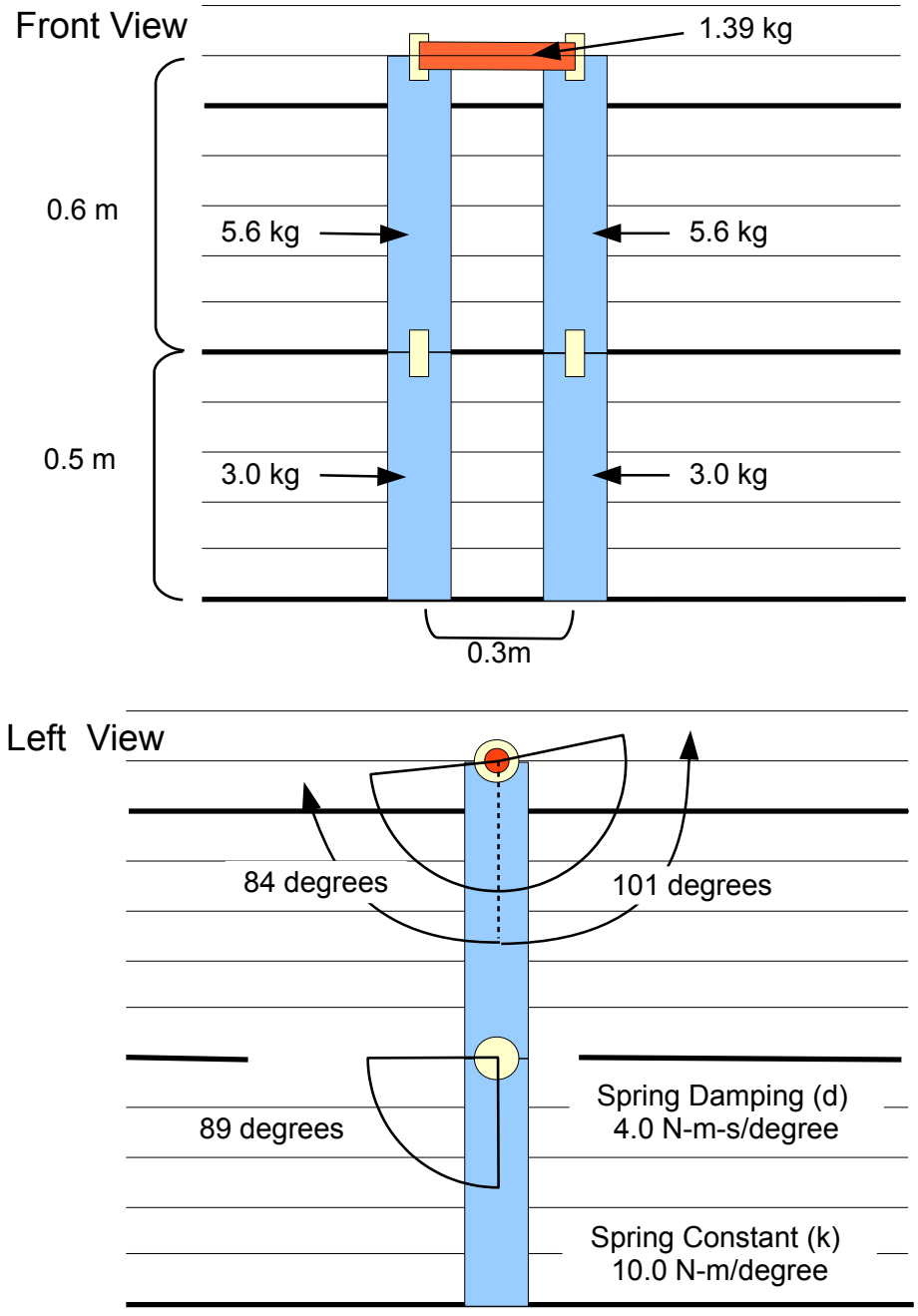


Figure 3.1: **Biped Physical Model:** dimensions, weights, joint angular limits and torsional spring constants.

choice of fitness measurement seeks to avoid assigning a high initial fitness value to those bipeds that simply take a large lunging step.

Outputs of the neural network nodes were passed through a low-pass filter before being mapped to the torsional angular displacement signal. This was to prevent the biped from moving simply by vibrating its feet at high frequency (this can result in motion similar to that of a washing machine when it enters its spin cycle and starts moving across the floor).

Each time-step for the neural network update lasts 0.025 seconds (or 1/40th of a second). Thus a 20 second trial corresponds to 800 neural network time-steps. The physical simulation uses forward Euler integration with a time-step of 50 micro-seconds (i.e. there are 500 dynamics time-steps for each neural network time-step). The small physical time-step was necessary to ensure the stability of the physical simulation.

The trial was terminated if the biped fell below 50 percent of its original height and a fitness value assigned based on the distance travelled up to that point in time.

3.2.1.4 Sensorimotor System

Motor output signals are taken from the 1st 5th, 9th, and 13th nodes in the network . These nodes were used to control the rest position of the torsional springs in the biped's left hip, right hip, left knee and right knee respectively. The output of these nodes was mapped to a range of -94 to 101 degrees angular hip displacement, and 0 to -89 degree angular knee displacement.

There are two sets of sensor configurations used in this experiment (nominally referred to as R and I configurations). The majority of simulations were carried out with a regular connection pattern (R configuration). Sensor input from the right foot contact sensor occurs at the 2nd, 6th, 10th and 14th nodes. Sensor input from the left foot contact sensor occurs at the 3rd, 7th, 11th and 15th nodes. Sensor input consisted of a binary 1.0 or 0.0 value depending upon whether or not the feet of the biped were in contact with the ground.

Additional simulations were carried out on GasNet networks with an irregular sensor connection pattern (I configuration). In this case sensor signals were input

to the 8th, 9th and 10th nodes for the right foot, and the 11th, 12th, 13th and 14th nodes for the left foot. One of the reasons for this second sensor configuration was to see if there were any significant changes in the performance of circuits where sensor nodes were connected directly to motor nodes (i.e. in the case of the 9th and 13th nodes in the I configuration). These node assignemnets are summarized in tables 3.1 and 3.2.

Table 3.1: Biped Motor Nodes

Joint	Motor Nodes
Left Hip	1
Right Hip	5
Left Knee	9
Right Knee	13

Table 3.2: Ground Contact Sensor Nodes

Body	Sensor Nodes Regular(R)	Sensor Nodes Irregular (I)
Left Foot	3,7,11,15	11,12,13,14
Right Foot	2,6,10,14	8,9,10

3.2.2 Network Variants and Characteristic Equations

This section describes the network details of each of the varieties of networks tested. The focus is on the characteristic equations that govern the dynamics of each network variety. Equation notation most closely resembles that used by Beer (Beer [1995b]). In some cases it is necessary to deviate from the original author's notation so as to try and maintain a consistent notation over different network varieties.

A list of the network types is shown in Table 3.3. Each network type is assigned an index for reference purposes numbered between 1 and 14. The R letter denotes the regular sensor configuration; the I denotes the irregular sensor configuration. Each network type is assigned an arbitrary index for reference purposes.

It should be noted that conventional GasNets comprise nodes that are spatially distributed. A parametric coding strategy is used where connections are

Table 3.3: Summary of Network Types

Type	Description
1R	Conventional CTRNN
2R	Center-Crossing CTRNN
3R	Basic PNN
4R	Gas-Modulated PNN
5R	CTRNN/PNN Hybrid
6R	Conventional GasNet
7R	Fully Recurrent GasNet
8R	CTRNN/GasNet Hybrid no Gas
9R	CTRNN/GasNet Hybrid with Gas
10I	Conventional GasNet 16 Cell
11I	Conventional GasNet 32 Cell
12I	Fully Recurrent GasNet
13I	CTRNN/GasNet Hybrid no Gas
14I	CTRNN/GasNet Hybrid with Gas

determined for each node based on genetic parameters that define geometric arcs originating at each node. A node that falls within an excitatory or inhibitory arc is deemed to be electrically connected to the node at origin of the arc (see Husbands [1998a] for specific details). The consequence of this, is that conventional GasNets are only sparsely connected. This approach is used in the the following network types; 6R, 8R, 9R, 13I and 14I. All other network types are fully interconnected.

3.2.2.1 Conventional CTRNN - type 1R

This is a conventional CTRNN based on the work of (Beer [1995b]). Node self-recurrency is allowed. Node connection weights and bias are under evolutionary control.

$$y_i^{t+1} = y_i^t + \frac{T}{\tau_i}(-y_i^t + \sum_{j=1}^{N} \omega_{ji}\sigma(y_j^t + \theta_j) + I_i) \tag{3.19}$$

$$i = 1, 2, ..., N$$

Where:

y_i^{t+1} is the activation of the i'th node at time $t+1$.

y_i^t is the activation of the i'th node at time t.

τ_i is the time constant for the i'th node calculated according to equation equation 1.

I_i a sensor input to the i'th node where I is either 1 (in contact with the floor) or 0 (not in contact with the floor).

θ_j a bias term for the j'th node where $\theta\epsilon[-2, 2]$.

T is the time slice (in this case T is set to 1).

ω_{ji} is the weight of the output from the j'th node to the i'th node where $\omega\epsilon[-4.0, 4.0]$.

σ is the logistic activation function.

$$\sigma(z) = \frac{1}{(1 + e^{-z})} \tag{3.20}$$

3.2.2.2 Center-Crossing CTRNN - type 2R

The characteristic equation of the Center-Crossing CTRNN from Mathayomchan and Beer [2002a] is the same as that of the CTRNN (type 1). However initial biases are calculated such that:

$$\theta_i = \frac{-\sum_{j=1}^{N} \omega_{ji}}{2} \tag{3.21}$$

These authors suggest that populations seeded with center-crossing networks may be more likely to yield a wider range of dynamics than a population of random networks.

3.2.2.3 Basic PNN - type 3R

This implementation is based on the description given by Urzelai and Floreano (Urzelai and Floreano [2000a]). The key characteristic of PNN's is that connection

weights vary over time based on Hebbian learning rules given by:

$$\omega_{ji}^{t} = \omega_{ji}^{t-1} + \eta\Delta\omega_{ji} \tag{3.22}$$

Where η is a learning rate ($0.0 < \eta < 1.0$) and ω_{ji} is the connection weight of the input to node i from node j. The adaptation rule $\Delta\omega_{ji}$ is genetically determined for each node. All inputs to a given node are subject to the same adaptation rule (referred to as node encoding by the authors). Whilst equation 3.22 denotes the general form of the equation used to determine weight changes, the specific form of the adaptation rule $\Delta\omega_{ji}$ for each node is one of four types described below.

Where x is the activation of node j, which is an input to node i (which has an output activation of y), the adaptation rule is one of:

Plain Hebbian Rule

$$\Delta\omega_{ji} = (1 - \omega_{ji})x_j y_i \tag{3.23}$$

Post-Synaptic Rule

$$\Delta\omega_{ji} = \omega_{ji}(-1 + x_j)y_i + (1 - \omega_{ij})x_j y_i \tag{3.24}$$

Pre-Synaptic Rule

$$\Delta\omega_{ji} = \omega_{ji}(-1 + y_i) + (1 - \omega_{ji})x_j y_i \tag{3.25}$$

Covariance Rule

$$\Delta\omega_{ji} = \begin{cases} (1 - \omega_{ji}) & \text{if } F(x_j, y_i) > 0 \\ (\omega_{ji})F(x_j, y_i) & \text{otherwise} \end{cases} \tag{3.26}$$

Where:

$$F(x_j, y_i) = \tanh(4(1 - |x_i - y_j| - 2) \tag{3.27}$$

All nodes in the PNN are fully interconnected (with self-connections also supported). The rate of learning η can only assume one of four values (0.0, 0.3, 0.6, 0.9). The characteristic equation for the PNN is shown below:

$$y_i^{t+1} = \sigma(\sum_{j=1}^{N} \omega_{ji}^{t}(y_j^t)) + I_i \quad i = 1, 2, ..., N \tag{3.28}$$

Where:

ω_{ji}^t is the adaptive weight for the j'th input to the i'th node.

σ is the standard logistic activation function.

I_i a sensor input to the i'th node where I is either 1 (in contact with the floor) or 0 (not in contact with the floor).

The term "basic" is used to differentiate it from that used by Tuci and Quinn (Tuci and Quinn [2003]) where an additional bias input was used for each node. In common with the implementation described by Blynel and Floreano (Blynel and Floreano [2002]) the range of y_i is $[0, 2]$ for input neurons and $[0, 1]$ for hidden and output neurons.

3.2.2.4 Gas-Modulated PNN - type 4R

This is essentially the same as the basic PNN (type 3) with the exception that nodes whose weights are genetically determined to be modified by the Plain Hebbian Rule, or Post-Synaptic Rule, have their weights modified by diffused gases (the gas diffusion is described in Husbands et al. [1998b]). For these two varieties of nodes, the weight modification rule becomes:

Gas Modified Plain Hebbian Rule

$$\Delta\omega_{ji} = (\frac{c_{1i}^t}{c_{1i}^t + c_{2i}^t})(1 - \omega_{ji})x_j y_i \tag{3.29}$$

Gas Modified Post-Synaptic Rule

$$\Delta\omega_{ji} = (\frac{c_{2i}^t}{c_{1i}^t + c_{2i}^t})(\omega_{ji}(-1 + x_j)y_i + (1 - \omega_{ij})x_j y_i) \tag{3.30}$$

Where:

c_{1i}^t is the concentration of gas 1 at the i'th node.

c_{2i}^t is the concentration of gas 2 at the i'th node.

The significance of these equations is that the rate of change in the weights of the inputs to these nodes will vary continuously with changes in the relative concentration of these two gases. When both gases have zero concentrations there is no change in weight.

3.2.2.5 CTRNN/PNN Hybrid - type 5R

This is a modification of the conventional PNN, with activation signals modified by a node based time constant under evolutionary control (in a similar fashion to conventional CTRNNs).

$$y_i^{t+1} = y_i^t + \frac{T}{\tau_i}(-y_i^t + \sum_{j=1}^{N} \omega_{ji}^t \sigma(y_j^t + \theta_j) + I_i) \tag{3.31}$$

$$i = 1, 2, ..., N$$

Where:

ω_{ji}^t is the adaptive weight for the j'th input to the i'th node.

3.2.2.6 Standard GasNet - type 6R

The implementation in this report is based on GasNet work reported in (Husbands et al. [1998b]).

In GasNets, node transfer functions can be modulated by local gas concentrations in the vicinity of the node. Nodes can also act as chemical emitters, under either gas or electrical stimulation. GasNet nodes exist in a geometric plane where internode distances determine gas concentrations and (in conjunction with additional genetic parameters) network connectivity. Under typical evolutionary parameters the GasNet connectivity rules result in a sparsely connected network.

$$y_i^{t+1} = \tanh[k_i^t(\sum_{j \epsilon C_i} \omega_{ji} y_j^t + I_i) + b_i] \tag{3.32}$$

Where:

k_i^t is a time-varying transfer function modulator. The value of k varies with gas concentrations at the i'th node.

C_i is the set of all nodes that have an input to the i'th node.

I_i a sensor input to the i'th node.

b_i a bias term for the i'th node.

3.2.2.7 Fully Recurrent GasNet - type 7R

The fully recurrent GasNet uses the GasNet model of gas diffusion but adopts a fully recurrent connectivity model. Whereas a conventional GasNet is sparsely connected, this network is fully connected.

$$y_i^{t+1} = \tanh[k_i^t(\sum_{j=1}^{N} \omega_{ji}\ y_j^t + I_i) + b_i] \tag{3.33}$$

$$i = 1, 2, ..., N$$

3.2.2.8 CTRNN/GasNet Hybrid no Gas - type 8R

This is a variation of the conventional CTRNN. In this case inter-node connectivity is determined by the approach used in GasNets. It uses a sigmoid transfer function, with bias and node time constants under evolutionary control.

$$y_i^{t+1} = y_i^t + \frac{T}{\tau_i}(-y_i^t + \tanh[K_i(\sum_{j \epsilon C_i} \omega_{ji} y_j^t + I_i) + b_i]) \tag{3.34}$$

Where:

K_i is a transfer function constant.

T is the time slice constant.

τ_i is the time constant for the i'th node.

3.2.2.9 CTRNN/GasNet Hybrid with Gas - type 9R

This is another variation of a conventional CTRNN, but in this case it is more extensively modified along the lines of GasNets. The network nodes transfer function is gas modulated, and network connectivity is based on the GasNet model. What remains of the the original CTRNN is the node time constant.

$$y_i^{t+1} = y_i^t + \frac{T}{\tau_i}(-y_i^t + \tanh[k_i^t(\sum_{j \epsilon C_i} \omega_{ji}\ y_j^t + I_i) + b_i]) \tag{3.35}$$

Where:

k_i^t is a time-varying transfer function modulator. The value of k varies with gas concentrations at the i'th node.

3.2.2.10 Conventional GasNet 16 Cell - type 10I

This network is based on the Conventional GasNet (type 6R) but uses the irregular sensor configuration. Another difference is that bias values are set such that $b_i \epsilon [-4, 4]$.

3.2.2.11 Conventional GasNet 32 Cell - type 11I

Other than the number of cells in this network, all other parameters are the same as those of the 16 cell gas net (type 10I). Note that of all the networks tested, this is the only one that is comprised of 32 cells or nodes.

Given the nature of the GasNet connectivity algorithm, 32 cell GasNets are likely to be more highly inter-connected than 16 cell GasNets. The size of the plane remains constant, so a higher number of cells means a higher cell density. A larger number of cells will tend to fall within a given connection arc, thus resulting in a higher number of inter-cell connections per cell.

3.2.2.12 Fully Recurrent GasNet - type 12I

This network is based on the Fully Recurrent GasNet (type 7R) but uses the irregular sensor configuration. Another difference is that bias values are set such that $b_i \epsilon [-4, 4]$.

3.2.2.13 CTRNN/GasNet Hybrid no Gas - type 13I

This network is based on the CTRNN/GasNet Hybrid without Gas (type 9R) but uses the irregular sensor configuration. Another difference is that bias values are set such that $b_i \epsilon [-4, 4]$.

3.2.2.14 CTRNN/GasNet Hybrid with Gas - type 14I

This network is based on the CTRNN/GasNet Hybrid with Gas (type 9R) but uses the irregular sensor configuration. Another difference is that bias values are

set such that $b_i \epsilon [-4, 4]$.

3.2.3 Results

Table 3.5 shows the peak fitness of the best individual in the ten evolutionary runs for each network type. Table 3.6 shows the average peak fitness value across all runs for each network type. Table 3.7 gives some indication of the phenotypic variation of the peak fitness across all runs for each network type.

To put these fitness values in perspective, 2.5 meters is attainable through a fast walk in about 10 seconds (attained by the Conventional GasNet type 10I), and through a slow walk in about 20 seconds (attained by the Conventional GasNet type 6R). Bipeds that have travelled around 1 meter have typically taken around two steps. The worst performing Basic PNN typically only extended one leg, before slightly drawing the lagging leg forward.

Figure 1 shows three snapshots of the biped walking. This rendering is based on the results obtained from the 16 cell GasNet (type 10I).

3.2.3.1 Principal Results Summary

We can summarize the most important results as follows:

1. Conventional GasNets were the only networks to achieve cyclical bipedal locomotion.

2. Conventional GasNet peak fitness was typically more than twice that of the other networks.

3. Center-Crossing CTRNNs achieved the second highest peak fitness after the conventional GasNets.

4. Center-Crossing CTRNNs and CTRNNs attained the highest average fitness values.

5. Center-Crossing CTRNNs marginally outperformed conventional CTRNNs in peak and average fitness.

6. Basic PNNs without dynamic attributes performed the worst.

7. Both a CTRNN/PNN Hybrid and a Gas Modulated PNN improved on the performance of the Basic PNN.

8. The CTRNN/PNN Hybrid achieved comparable results to the Conventional CTRNN.

9. Fully Connected GasNets perform badly compared to Conventional GasNets. GasNet performance seems to decline with increased inter-connectivity (see type 11I vs 10I and 6R).

10. Conventional GasNets exhibit a higher variation in phenotype fitness than other network types. Conventional CTRNNs exhibit relatively low phenotypic variation.

11. Gas Modulated networks generally outperform their un-modulated counterparts (see type R8 vs R9 and I13 vs I14).

12. GasNet performance is largely unaltered by minor sensorimotor configuration changes (see R vs I sensor configurations).

These results are generally consistent with those reported by GasNet researchers and the comparative studies cited earlier.

3.2.3.2 The Fittest

The two fittest networks (type 10I and 6R) were both Conventional GasNets. Despite both achieving almost identical fitnesses, the phenotypic solutions were qualitatively different. Gases were actively oscillating in the 10I network, whereas they remained at constant levels in the 6R network. This is a strong indication of the ability of GasNets to discover alternative pathways to a given fitness level.

Whilst Figure 3.2 (type 10I) may appear to imply a simple oscillatory behaviour, analysis of network activity shows that biped motion is substantially reactive. The transition in foot contact sensor data from not-touching to touching (a 0.0 input value to a 1.0 input value at the sensor nodes) triggers a substantial change in network activity. The left foot in a raised position suddenly touching the ground will elicit a change in the control signal applied to the right hip, such that it begins to rise.

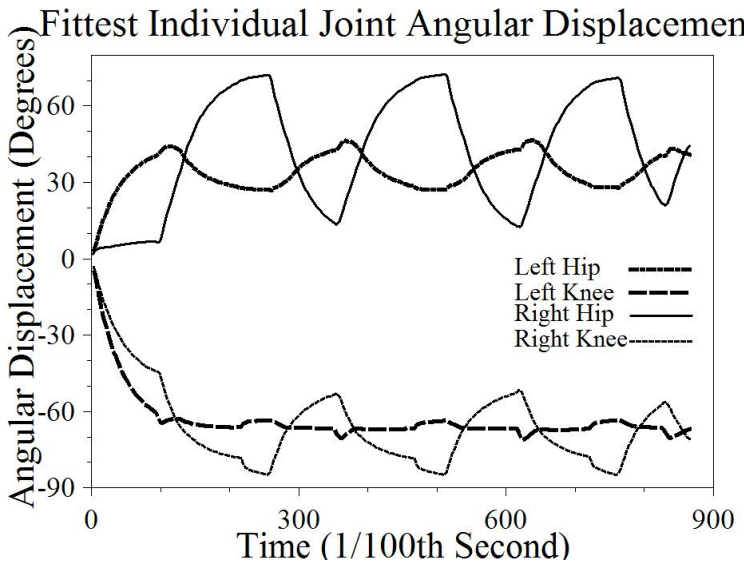


Figure 3.2: GasNet Biped Kinematic Data: (Upper) Left Hip and Right Hip (Lower) Right Knee and Left Knee

Table 3.4: Combined Results

Network	Rank	Max.	Avg	STD
6R Conventional GasNet	1	2.63	0.97	0.70
10I Conventional GasNet 16 Cell	2	2.62	0.92	0.66
2R Center-Crossing CTRNN	3	1.63	1.13	0.24
11I Conventional GasNet 32 Cell	4	1.31	0.94	0.29
5R CTRNN/PNN Hybrid	5	1.27	0.90	0.33
14I CTRNN/GasNet Hybrid with Gas	6	1.25	0.92	0.21
1R Conventional CTRNN	7	1.24	1.11	0.15
9R CTRNN/GasNet Hybrid with Gas	8	1.22	0.95	0.25
8R CTRNN/GasNet Hybrid no Gas	9	1.18	0.90	0.23
4R Gas-Modulated PNN	10	1.06	0.57	0.30
7R Fully Recurrent GasNet	11	0.98	0.35	0.09
13I CTRNN/GasNet Hybrid no Gas	12	0.72	0.61	0.13
12I Fully Recurrent GasNet	13	0.53	0.47	0.21
3R Basic PNN	14	0.24	0.23	0.01

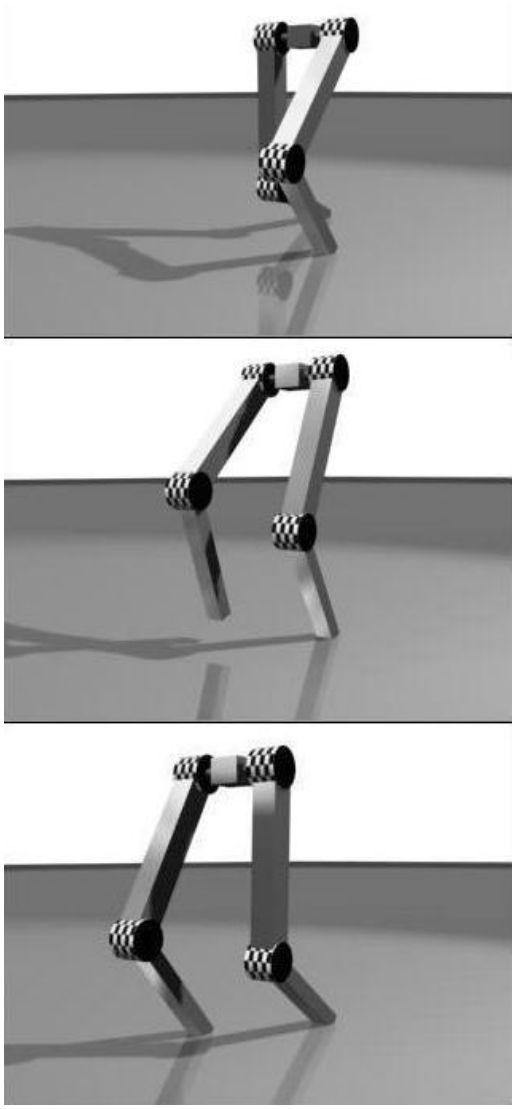


Figure 3.3: Renderings of the fittest GasNet biped at 5.0, 5.875 and 6.25 seconds.

Table 3.5: Distance Travelled by the Fittest Individual (meters).

Network	Rank	m
6R Conventional GasNet	1	2.63
10I Conventional GasNet 16 Cell	2	2.62
2R Center-Crossing CTRNN	3	1.63
11I Conventional GasNet 32 Cell	4	1.31
5R CTRNN/PNN Hybrid	5	1.27
14I CTRNN/GasNet Hybrid with Gas	6	1.25
1R Conventional CTRNN	7	1.24
9R CTRNN/GasNet Hybrid with Gas	8	1.22
8R CTRNN/GasNet Hybrid no Gas	9	1.18
4R Gas-Modulated PNN	10	1.06
7R Fully Recurrent GasNet	11	0.98
13I CTRNN/GasNet Hybrid no Gas	12	0.72
12I Fully Recurrent GasNet	13	0.53
3R Basic PNN	14	0.24

Table 3.6: Average of Distances Traveled by the Fittest individual in each of the ten runs (meters).

Network	Rank	m
2R Center-Crossing CTRNN	1	1.13
1R Conventional CTRNN	2	1.11
6R Conventional GasNet	3	0.97
9R CTRNN/GasNet Hybrid with Gas	4	0.95
11I Conventional GasNet 32 Cell	5	0.94
14I CTRNN/GasNet Hybrid with Gas	6	0.92
10I Conventional GasNet 16 Cell	7	0.92
5R CTRNN/PNN Hybrid	8	0.90
8R CTRNN/GasNet Hybrid no Gas	9	0.90
13I CTRNN/GasNet Hybrid no Gas	10	0.61
4R Gas-Modulated PNN	11	0.57
12I Fully Recurrent GasNet	12	0.47
7R Fully Recurrent GasNet	13	0.35
3R Basic PNN	14	0.23

Inspection of the graph shows that the gait is asymmetric. The right hip oscillates between approximately 15 and 73 degrees, whereas the left hip oscillates between 26 and 47 degrees. The lower left leg remains at an almost constant angular displacement to the upper leg, whereas the right knee-joint oscillates in an anti-phase relationship with the right hip-joint.

3.3 Experiment 2: Quadrupedal Locomotion

3.3.1 Introduction

3.3.2 Network Descriptions

Put in the simplest terms, Continuous Time Recurrent Neural Networks (CTRNNs) (Yamauchi and Beer [1994a]) represent the "plain vanilla" form of DRNN's, GasNets represent an approach to incorporate neuromodulation into a form of DRNN (Husbands et al. [1998b]), and Plastic Neural Networks (PNNs) seek to incorporate Hebbian dynamics (Floreano and Mondada [1996]). The particular variants used in this experiment are described in detail in this following section.

One thing that should be noted is that for each network, network morphology has been constrained to correspond more closely to that associated with coupled-oscillator circuitry. All networks comprise a total of 16 nodes or cells. An initial population is seeded with networks that have a single symmetry axis, such that we have two subnetworks of 8 nodes, each with identical parameters. In the case of GasNets (where nodes have a physical location in a 2 dimensional plane) the position of each node from one subnetwork is mirrored in an axis that divides the plane. This is shown more clearly in Figure 3.4. The mirrored nodes are interconnected via mutually inhibitory connections. Whilst initial populations comprise symmetrical networks, mutation and crossover results in the introduction of asymmetries over a period of time (symmetry is only enforced in the initial population).

Table 3.7: Standard Deviation of the distance travelled by the fittest individual in each of the ten runs.

Network	Rank	SD
6R Conventional GasNet	1	0.70
10I Conventional GasNet 16 Cell	2	0.66
5R CTRNN/PNN Hybrid	3	0.33
4R Gas-Modulated PNN	4	0.30
11I Conventional GasNet 32 Cell	5	0.29
9R CTRNN/GasNet Hybrid with Gas	6	0.25
2R Center-Crossing CTRNN	7	0.24
8R CTRNN/GasNet Hybrid no Gas	8	0.23
14I CTRNN/GasNet Hybrid with Gas	9	0.21
12I Fully Recurrent GasNet	10	0.21
1R Conventional CTRNN	11	0.15
13I CTRNN/GasNet Hybrid no Gas	12	0.13
7R Fully Recurrent GasNet	13	0.09
3R Basic PNN	14	0.01

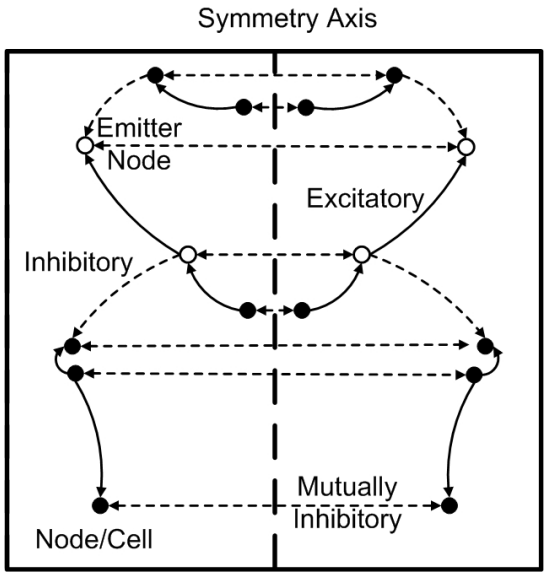


Figure 3.4: Schematic of the symmetrical distribution of GasNet nodes

3.3.3 Experimental Setup

A screen-shot of the quadruped used in this experiment is shown in Figure 3.5. Whereas the previous (bipedal locomotion) study used a physics package called AutoSim, this study uses an open source package called Open Dynamics Engine (ODE).

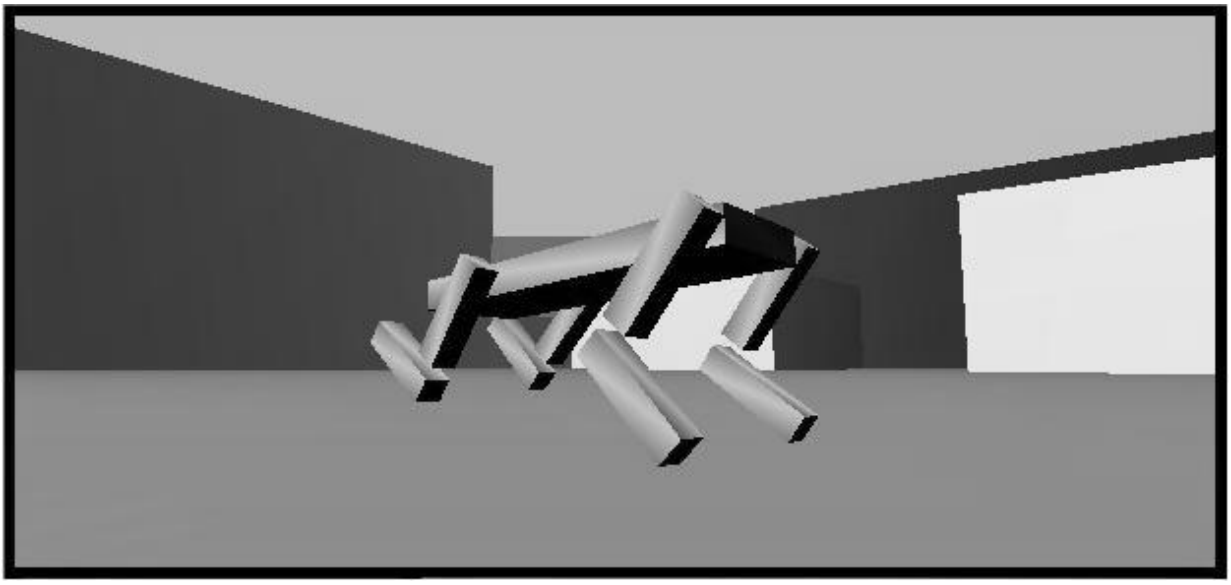


Figure 3.5: Physically Simulated Quadruped

The quadruped torso is simulated with 6 physical degrees of freedom (unlike the previous biped study, where the biped was physically incapable of falling sideways). The quadruped comprises 9 rigid bodies, two rigid bodies for each leg, and a single rigid body for the quadruped torso. Lower limbs are connected to the upper limbs via a limited hinge joint with a single rotational degree of freedom. Upper limbs are connected to the torso again with a limited hinge joint with one degree of freedom. The angular limits are shown in a scale diagram of the quadruped in Figure 3.6.

The assessed fitness of each individual is simply taken as the absolute distance travelled by the quadruped in a fixed time interval. The neural network is updated at half the frequency of the physics simulation, for a total of 5000 updates (approximately 20 seconds of real-time simulation).

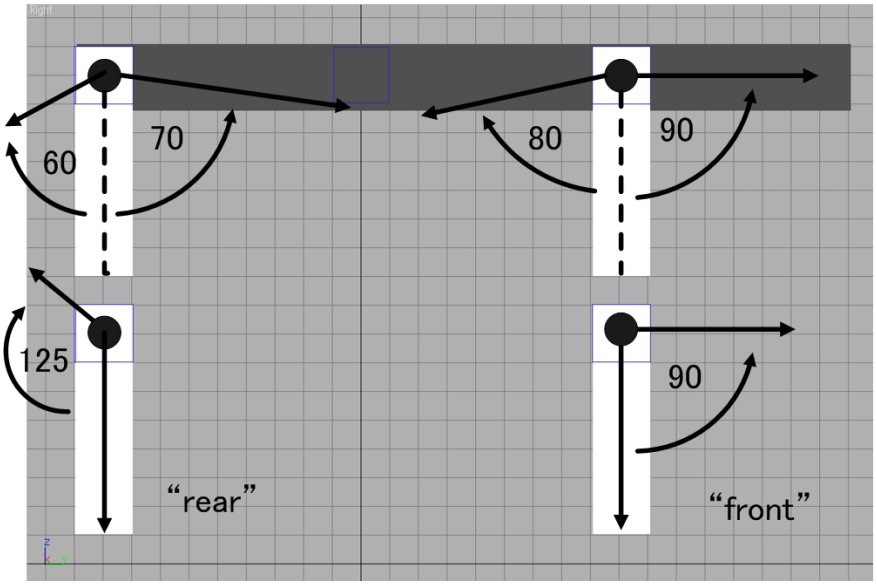


Figure 3.6: Scale drawn diagram showing joint angular limits (in degrees)

Table 3.8: Sensor Nodes

Limb	Sensor Nodes
Right Rear	0,1
Left Front	5,6
Left Rear	8,9
Right Front	12,13

Sensor input to the neural network comprises simple contact sensors associated with each lower limb. When a lower limb is in contact with the ground, the sensor value is 1. At all other times it is zero. Each contact sensor is connected to two network nodes as shown in Table 3.8.

Motor output nodes are shown in Table 3.9. The output signal of each motor node is mapped linearly into the hinge angular range. This becomes a target angular displacement. A velocity value for the joint is then calculated, based on the difference between the current angular displacement and this target displacement. The physics engine than applies torque necessary to arrive at this joint velocity, constrained by a maximum torque value.

Table 3.9: Motor Nodes

Joint	Motor Nodes
Hip Right Rear	0
Knee Right Rear	2
Hip Left Front	4
Knee Left Front	6
Hip Left Rear	8
Knee Left Rear	10
Hip Right Front	12
Knee Right Front	14

3.3.4 Results

The results of each evolutionary run are shown in Table 3.10. The distance travelled is normalized by the body length of the quadruped so as to present the data in a more intuitive fashion. A distance travelled of 1.3 body lengths simply corresponds to the quadruped falling forwards. Between 2 and 3 body lengths, typically one or two steps have been taken. Distances greater than 4 body lengths usually correspond to a slow or unstable gait. Distances greater than this correspond typically to cyclical gaits. The global fitness peak is likely to be around 14 body lengths.

Although the results for GasNets and the hybrid CTRNN/PNN are very similar, there are differences in the stability of evolved gaits, with those of the GasNets exhibiting greater stability. Over all, the fittest individual was evolved using GasNets, however the CTRNN/PNN achieved a marginally higher average fitness measure. The results for the Center-Crossing CTRNN were generally poor. However, one of the runs did discover the same gait as the fittest GasNet and CTRNN/PNN. As a consequence the fittest Center Crossing CTRNN individual attained a fitness very close to that of the other two networks considered.

3.3.5 Discussion

Although not obvious from Table 3.10, the quality of motion exhibited by the quadruped varied substantially with different networks. The fittest CTRNN driven quadruped exhibits motion that is similar to what we might expect from

Table 3.10: Distance Travelled by the Fittest Individual (normalized to quadruped body length). Letter superscripts correspond to distinct gaits described in the text.

Rank	CTRNN	GasNet	CTRNN/PNN
1	11.7^a	13.7^a	13.6^a
2	1.5	7.2^b	7.2^f
3	1.4	5.2^c	6.2^e
4	1.4	4.6	6.2
5	1.4	3.4^d	5.7^e
6	1.4	3.2	4.9
7	1.4	3.0	4.2
8	1.4	2.7	1.6
9	1.3	2.6	1.5
10	1.3	2.5	1.3
11	1.3	2.0	1.3
12	1.3	2.0	1.3
13	1.3	2.0	1.3
14	1.3	1.7	1.1
Average.	2.1	4.0	4.1
Median.	1.4	2.8	2.9
Maximum.	11.7	13.7	13.6

a Central Pattern Generator. The gait is symmetrical and the frequency of oscillation appears to be relatively stable. The CTRNN produces a stable gait that continues for a prolonged time period after the end of a trial (if allowed to continue).

In contrast, the gaits generated by the CTRNN/PNN hybrid appear to be highly reactive, with little evidence of forced oscillations (excluding that of the fittest CTRNN/PNN individual). Although the order of stepping may assume a regular pattern, there is considerable variability in the speed of subsequent steps. In this respect motion closely resembles irregular passive dynamic walking. The GasNet demonstrates gaits which exhibit aspects of reactive behavior together with forced oscillations (producing the fasted quadruped within the evaluation time period).

The GasNet and CTRNN/PNN Hybrid exhibited the widest range of gaits. The CTRNN/PNN exhibited some gaits that were not discovered by GasNet,

although the CTRNN/PNN gaits were relatively unstable. If we consider the form of the CTRNN/PNN network, it is clear that connection weights will gradually decline if there is a lack of coincident activity. In such a dynamic environment, rich external sensory input may play a more significant role than it would in networks that exhibit strong intrinsic dynamic activity (such as self-oscillation). This may well go some way to explaining why the gaits exhibited by the CTRNN/PNN appear to be more reactive, but seem to lack strong oscillatory activity.

The evolved quadrupeds exhibit a variety of of the gaits, and body configurations. The quality of the motion varies from driven-oscillatory to ballistic-reactive. Some of the most distinctive patterns are described below (the letters correspond to that which appears next to the fitness value in Table 3.10).

a The front legs hit the ground together, then the back legs, corresponding to the bound gait.

b The quadruped jumps from its rear limbs, stopping itself with its fore-limbs, before returning to a squatting position. This cycle then repeats. This does not correspond to any of the commonly observed animal gaits.

c In this case the left fore-limb remains in a forward position, whilst the right fore-limb remains in a rearward position. Rear limbs push off from the ground in a coordinated fashion. Motion resembles that of the three-legged bound gait.

d In some runs, the quadruped assumes a crawling configuration. Early in the evolutionary run a suspended walking gait is evident. Two diametrically opposed limbs are always in contact with the ground.

e This motion pattern most closely resembles ballistic walking. It is a highly irregular gait, with little evidence of regular oscillatory movement.

f This is another bounding gait. However it makes use of "elbows" rather than "hands/feet" in its fore-limbs.

3.4 Summary

3.4.1 An Idealised Comparative Study

An ideal comparative study would resemble an ecosystem of creatures in their environment. They would compete amongst themselves, and compete with other organisms in pursuit of survival and themselves and their offspring. They would discover for themselves what activities were necessary for survival through the process of dynamic or continuous learning in addition to undergoing genetic and ontogenetic adaptation. An environment of this complexity could conceivably be constructed, and its architecture would resemble that of a massive multiplayer on-line role playing game (MMORG), since it would require both distributed computation and distributed simulation (i.e. the simulation environment is common to client machines). Unlike a MMORG there would be no requirement for human intervention. In such a simulation, comparable neural networks could be evaluated as separate species, and their population would prosper or decline in as a consequence of the evolutionary processes embedded in the simulation. In such an idealized study, the dimensions of comparison highlighted in section 3.1.2 would be parametrized, and would themselves be the subject of evolutionary search. In is conceivable that such studies could be carried out in the future, although not without the significant concentration of software development efforts on a single project.

The studies described here represent the limits of what is practically achievable by working in isolation over a prolonged period by a single researcher. This involves a sampling of neural network architectures based on a matrix of neural network parameters with dimensions of; connection density, neural density (in the case of spatial models such as the Gasnet), alternative functional models (GasNet, Hebbian networks and continuous time recurrent neural networks), hybrid variations of these functional models, and morphology (in this case only bipedal and quadrupedal). In considering the contribution of this work, it is important to remember two things; firstly that at the time that this study was conducted no one had carried out any comparative studies of this scope in evolutionary robotics and secondly that the functional models compared represented

the most sophisticated models existent at that time (excluding those employing ontogenetic models).

A factor completely ignored in these studies, is the impact that the use of alternative *genetic algorithms* might have on the relative performance of these networks. The efficiency with which we are able to explore phenotype space, is a complex function of the encoding strategy, ontogenetic function (where relevant in the case of indirect encoding), recombination, and mutation operators. What we have done in this comparative study is to compare the functional characteristics of the evolved neural networks, and implicitly the coding strategies and ontogenetic functions associated with each network model. What we have not done, is to explore the impact of alternative mutation and recombination operators on the evolvability of each network form. An exploration of these dimensions would require a further significant increase in the requirement for computational resources. An ideal study would however also incorporate an investigation of effects of variations in these evolutionary algorithm operators.

There is a limit to that which it is worthwhile exploring at this stage. These networks are applied only to one problem domain, that of locomotion. Since our stated goal is of discovering techniques to extend the bounds of behavioural complexity that might be simulated, there is a limit to how much time we should focus on arriving at an optimal formulation for locomotion. Such a formulation may or may not integrate well with other neural networks having complementary functional characteristics. What we require at this stage is a rough measure of the relative performance of available network formulations for locomotion. Based on this, we can start to consider further lateral investigations involving alternative problem domains.

3.4.2 Discussion

3.4.2.1 Temporal Adaptivity and Evolvability

It is claimed that GasNets have a high evolvability due to their high capacity for *temporal adaptation* (Smith et al. [2002b]). The coupling of electrical and gas dynamics is discussed in some detail by Philippedes et al (Philippedes et al. [2002]). In this work the authors showed that reducing the likelihood that a node

pair are both chemically and electrically coupled can improve the evolvability of the GasNet network even further. In particular they state;

> .. systems involving distinct yet coupled processes are highly evolvable when there is a bias towards loose coupling between the processes; this allows the possibility of 'tuning' one against the other without destructive interference.

The relatively poor performance of fully connected GasNets, where a high degree of coupling is forced into the network, tends to support this claim. Philippides et al. also refer to the multiple redundancies inherent in loosely coupled GasNets which potentially lead to increased numbers of routes through the evolutionary search space. These factors may help to account for the relatively high variance in GasNet fitness in comparison with CTRNNs, as well as the significantly higher peak fitness. The inherent dynamics are being shaped and explored in a very different way.

In the case of the one of the fittest GasNet bipeds (type I10) the reactive response of the right hip to the transition from non-contact to contact of the left foot appears to be primarily gas-mediated. In contrast the joint angular displacement dynamics are governed primarily by electrical activation signals. Whilst this may be an over-simplification (GasNets are integrated systems with co-dependencies between both gas and electrical signalling), it is easy to imagine circumstances in which there are independent phenotypic processes that have intrinsic time dynamics associated with them. The ability to explore these phenotypic temporal dynamics in parallel may be a significant factor in the evolvability of GasNets.

Concepts such as of temporal adaptivity and system coupling may be useful concepts in helping us to understanding the dynamics of such systems. However we still need to try and identify the specific characteristics of these networks that support temporal adaptivity. Two of the most obvious areas to consider are the frequency and phase characteristics of the GasNet.

3.4.2.2 Phase Space Exploration

Phase relationships between signals in an articulated agent are extremely important to achieve coordinated activity. One aspect of GasNets is that nodes are

physically distributed in a virtual 2-dimensional space. A variation in the distance between nodes (through mutation for example) results in a phase lag or lead in the modulation of gas-coupled nodes. In this sense phase relationships are under direct evolutionary control. None of the other networks investigated in this study embody these characteristics. Simply put, in GasNets, node position mutations are operating directly in phenotype phase space (where nodes are chemically coupled).

In direct encoding, if a single parameter maps onto a specific phenotypic attribute that is largely independent of other attributes, this is likely to aid the efficient exploration of phenotypic space via mutation operators, since it may result in a smoother fitness landscape. In the case of articulated bipedal locomotion, we might imagine that the phase relationship between hip joints is a significant phenotypic attribute that affects overall fitness. A network model that can explore this phase relationship through mutation operators may exhibit greater temporal adaptation than one that cannot.

3.4.2.3 Frequency Space Exploration

Examination of the motor output signals of GasNets compared to other networks indicates that GasNets generate motor control signals with a wide range of frequency components (particularly high frequency). It is easy to imagine that a network that exhibits significant oscillatory behaviour over a wide bandwidth may have some advantage in seeking to discover a solution suitable for articulated motion.

In an attempt to analyze this further, five additional simulations were carried out based on the Conventional CTRNN (type 1R). In this case neural net updates took place at five times the frequency of the the tests described here (4000 neural net time steps in a 20 second period instead of 800). Peak fitness attained was 1.30 and average fitness was 1.19. These values are very similar to those of a CTRNN operating at a the lower frequency. This may imply that it is not just frequency ranges that are significant, but also the diversity of signal frequencies intrinsically present in the network, that help to determine temporal adaptivity.

3.4.2.4 Temporal Dynamics and PNNs

It should be stated that there is no reason to expect that the Basic PNN should have performed particularly well in this problem domain. The environment remains constant over evolutionary time. As such there is no additional benefit to be gained by in-trial learning. Limited sensor stimulation may have resulted in a rapid decay in node activity in this implementation. However, the modification of the Basic PNN to incorporate richer time dynamics (e.g. type 4R and 5R networks) can improve the performance of PNN networks.

3.4.3 Biped versus Quadruped Results

In the previous study, which considered bipedal locomotion, GasNets appeared to offer the best solution (the only network to achieve cyclical bipedal locomotion) followed by Center-Crossing CTRNNs. For detailed analysis of GasNet dynamics and performance, the reader is referred to (Smith et al. [2003a]) and (Philippedes et al. [2002]). In this respect the results for the GasNet are broadly in line with those of the previous study.

In this study, only one of the CTRNN runs resulted in locomotion. This is in keeping with prior work (Reil and Husbands [2002]) where CTRNNs were evolved for bipedal locomotion control. Only 10% of runs generated oscillatory activity that resulted in bipedal locomotion. No stable cyclical gaits were generated for the biped in the previous comparative study using CTRNNs.

Compared to the previous study, the biggest difference is in the relative performance of the Hybrid CTRNN/PNN. Whilst at best mediocre in the prior study, the results when applied to quadrupedal locomotion are comparable with those of the GasNet. There are two possible explanations that spring to mind. Firstly, modifying the original Hybrid CTRNN/PNN, so that it more closely resembles a coupled-oscillator, may result in dynamic activity that is more suitable to oscillation and locomotive control. Note that in the previous study networks were single heterogeneous networks with no axis of symmetry. Secondly, quadrupedal locomotion may be more amenable to reactive solutions than the intrinsically less stable problem of bipedal locomotion.

CPGs are currently the dominant motor control paradigm. Work has shown

that it is possible to model all the common quadrupedal gaits using a network of eight cells (Buono and Golubitsky [2001]). The results described here may lead us to question whether or not reactive responses are just as important in generating locomotive activity.

3.4.4 Conclusion

The value of a comparative study such as this is that it enables us; to verify the models of other researchers, determine whether or not there are trivial hybrid variants that might be superior to those models proposed, and to explore some of the design dimensions of these networks to determine if there are some principal factors that are the most significant. The variants that were tested represent a very sparse sampling of the total space of possible genotypes. However, doing substantially more than this would have required significantly greater computational resources (typically 12 computers were used for approximately 6 months to obtain these results).

An important factor to keep in mind when considering these comparative studies is that we are only addressing one problem domain, that of legged locomotion. The comparatives studies shown here indicate for the problem domain that we investigated, that the standard GasNet model represents a good solution (amongst those tested), although Hebbian (or PNN) networks also show comparable results in some cases. Since our goal is to establish how behavioural complexity can be best supported, it is sufficient for our purposes to establish that this is the case, since our goal is to investigate a number of problem domains. We are less interested in the optimal solution to locomotion, than the best approach to evolving networks capable of addressing a number of problem domains, and through their composite solution support behavioural complexity. There is not need necessarily at this stage to address locomotion further. An obvious question to ask at this stage is, 'are Gasnets suitable for other problem domains?'. Work described in the chapter 4 seeks to answer this question.

3.4.5 Subsequent and Recent Developments

The work conducted in this chapter was published in conference proceedings in 2004 (McHale and Husbands [2004c] and McHale and Husbands [2004b]). Since then there are been a number of developments in this field. We might best break these down into three seperate areas. The first relates to work that increases our understanding of the factors that determine the evolvability and performance in GasNets compared with other approaches (see for example Philippides et al. [2005], Magg and Philippides [2006] and Husbands et al. [2010]). The second relates to a diversification of the techniques used to address the evolution of systems capable of supporting locomotion or minimally cognitive behaviour, through largely dynamic processes; an Artificial Homeostatic System (AHS) comprising Non-Spatial GasNet (NSGasNet) and Artificial Endocrine Systems (AES) (see Moioli et al. [2008b] and Moioli et al. [2008a]), systems capable of supporting minimal cognitive behaviours supported solely by chemical models (Dale and Husbands [2010]), a hormone diffusion based model (Hamann et al. [2010]), kuramoto phase-based Models (Moioli et al. [2010]), the Essential Variable Monitoring GasNet (Lowe et al. [2010]), work extending Urzelai's Plastic Neural Networks (Hoinville et al. [2011]), work extending Randall Beers work on CTRNNs such as Adaptive Centre Crossing CTRNN's (Campo and Santos [2010], Santos and Campo [2012]), and novel approaches based on chaotic dynamics (Shim and Husbands [2012]).

The investigations that help to explain GasNet features and evolvability together with the new model classes are all worthy of significant attention. Unfortunately the scope of this thesis sacrifices depth in this particular area for breadth. It is noticeable that of the references quoted above, many represent extensions of circuit classes that we have already explored in this chapter. Since this thesis attempts to outline circuit varieties that are intrinsically complementary (taking inspiration from biological systems), the work outlined in this chapter is sufficient for our purposes.

Chapter 4

Experiment: Simple Ball Collection Experiment

4.1 Introduction

The results of our work described in chapter 3 indicate that GasNets offer us a good solution for locomotion in legged robots. Our original intention here was to simply determine the extent to which its properties were suited for alternative problem domains. Work described in this chapter focusses on one particular experiment. This experiment involves the collection of balls by a robot in a three-dimensional physically simulated environment. Such a task is typically solved (in the video games industry for example) by constructing Finite State Machines (FSM) that decompose a task into smaller problems, that are by themselves relatively trivial to solve. We might imagine an 'intelligent' solution to this problem would be for the robot to wander until discovering a ball. To then orient towards the ball, and then to approach it for collection. Early results produced by GasNets were not promising. As a consequence, the focus of attention shifted. The new goal was to determine whether we could evolve more intelligent behaviour by penalizing energy inefficiency in the fitness function. Chapter 5 goes on to examine this problem domain in greater detail, and determine what kind of circuit is best suited to solving such a problem and by implication determining the characteristics of the problem that makes it less amenable to solution by the

GasNets.

Often in such simulations the energy constraints placed on a simulated organism are implicit, either due to the physical design of the robot and its power supply, or through parameters chosen for actuators in physical simulation. Typically there is no cost penalty associated with higher energy use in the completion of a task, nor benefit in engaging in energy efficient activity. As a consequence, the behaviour exhibited by the evolved robots is often not typical of behaviour exhibited by living organisms, which clearly do have energy budgets determined by their activities and metabolism. This problem is of interest to evolutionary roboticists for two reasons; clearly energy efficient robots have greater economic value than those that are wasteful, and it may be more difficult to evolve intelligent behaviour where energy supplies are relatively unconstrained.

4.2 Energy and Life

At the macro level Energy processes are of major importance in driving Ecological Systems (Jorgensen and Bendoricchio [2001]) whilst at the micro level, energy governs the biological reactions that support life (Haynie [2001]). Within this broad spectrum of work, Biophysics and Biomechanics are amongst the most relevant areas to scientists seeking to evolve life-like robots or artificially simulated creatures. Interesting examples of the application of Biophysics to predict animal behaviour include that by D. M. Gates, who uses energy based models to predict lizard activity patterns, and predator-prey relationships (Gates [2003]). In the field of Biomechanics the extensive works of R. McNeill Alexander are of particular relevance to scientists considering locomotion in living organisms from an energy perspective (Alexander [2003]).

There are three key questions that are of pragmatic interest to researchers in Evolutionary Robotics. These are:

1. How do energy constraints influence behaviour?

2. How can we generate appropriate locomotor control systems that are energy efficient?

3. Does the imposition of energy constraints make it easier or more difficult to evolve effective integrated sensor-motor control systems?

Quite clearly the questions raised above are very broad in their scope. This paper describes a starting point and a basic methodology from which it is hoped that we can start to address some of these issues in greater detail. The focus of this paper is on the effect of the imposition of energy constraints on the relative utilization of motor and sensor facilities in solving a simple task.

4.3 Experimental Setup

Previous work (Husbands et al. [1998c]) involved evolving a GasNet based neural network for a real and simulated Robot, with the goal of moving towards a triangle and ignoring a rectangle. The experiment reported here differs in its focus on energy efficiency rather than the successful execution of a simple sensor-motor task. The following section describes the experimental set-up, together with details on the GasNet implementation and the evolutionary algorithm used. Whilst the following section describes the key experimental parameters, additional parameters that relate to the generation of the GasNet morphology are discussed in more detail in the source cited above.

4.3.1 The Robot and Its Environment

A model was made of a 'toy' robot with minimal sensor and motor capacity. The robot exists in a physically simulated 3d world. Motion is achieved through the application of linear and rotational forces to the robot at its centre of mass. A GasNet (described in more detail below) comprising 16 nodes is evolved to provide the motor signals for the application of forces to the robot. Each node represents a neuronal cell. Four of these nodes act as motor neurons. The rotational torque is determined by the sum of two motor output neurons. Torque is applied to the robot around a vertical axis centred at the robots centre of mass. The linear force is the sum of two motor output neurons, oriented in the robots forward direction, and passing through the robots centre of mass. As a result, the robot

is capable of rotating clockwise and counter-clockwise, as well as moving forwards and backwards.

The robot has four sensors. These comprise raycasts into the physically simulated world. This data is minimally pre-processed before passing to two sensor neuron inputs. One sensor input simply registers whether or not an object has been hit. Any of the rays striking an object will result in an input value of +1 applied to the sensor neuron input, and -1 when no objects are detected. The second sensor neuron receives a value that corresponds to the average distance from the ray source to the detected object across all sensor rays when an object is within sensor range. If a sensor ray detects no object, then the distance measured by the robot's sensor rays is taken to be the maximum value of the raycast's sensor range. The average distance value for all rays is mapped to value within the range [+1,-1]. Effectively we have two sensor modalities; the first neuron will detect any objects within the robot's visual range; the second sensor neuron will be activated when the robot is close to a target object. The raycast sensors and robot can be seen in Figure 4.1.

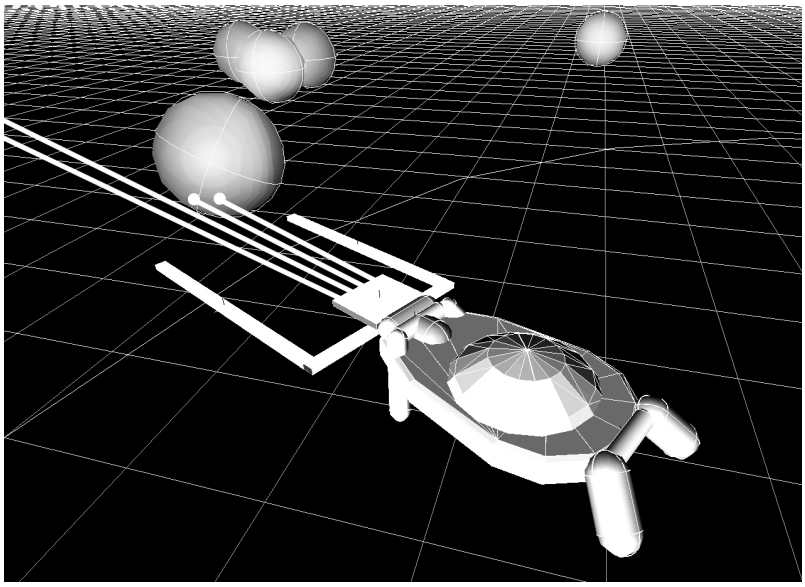


Figure 4.1: Side View of Robot and Sensor Rays

The robot exists in a planar world within which, in initial experiments, 24 large

spheres are placed. The jaws of the robot are slightly wider than the width of the spheres. The width of the robot jaws is 16 units; the spheres have a diameter of 15 units. The spheres are randomly distributed within a two-dimensional annulus. An additional exclusion corridor (70 units wide) is created, such that a robot travelling straight forward or backwards will not collide with any spheres. The length of the robot is approximately 48 units, the range of sensor rays 100 units, the inner radius of the annulus is 95 units and the outer radius 400 units. The aim of the task is to capture a sphere in the robot's jaws.

4.3.2 Trial Description and Fitness Function

Each trial consists of 10 sub-trials. At the start of each sub-trial the robot is placed at the origin (the centre of the annulus), and 24 spheres are randomly distributed within the annulus (with the exception of the exclusion corridor described above). Typically one or two spheres would be within sensor range if the robot were to rotate 360 degrees around its vertical axis. Due to the high variability in the distribution of spheres in each sub-trial, a large number of sub-trials are required to establish a representative fitness value. Figure 4.2 shows the distribution of spheres for a typical sub-trial.

There are two forms of the fitness function used in this experiment. The first form is used in tests where there is no cost penalty for using the maximum energy available to the robot. In this case the fitness value of the robot is taken to be the relative closeness of the sphere as measured by the robots sensor rays (measured as a fraction of the total ray distance) at the termination of a trial. The trial is terminated if the robot closes to within 10 percent of its total ray distance (i.e. 10 units from the origin of the sensor rays). This corresponds to the sphere entering completely into the robot's jaws (which constitute a channel that is 20 units deep). Each sub-trial is weighted to be a 10th of the total fitness sum of the trial. The maximum theoretical fitness of the robot is 1.0, corresponding to 10 trials where the robot acquires a sphere target in its jaws. However, due to the early termination condition, practically the maximum fitness attainable is 0.9 plus a small value corresponding to the distance travelled in one physically simulated time-step, prior to the early termination condition being detected.

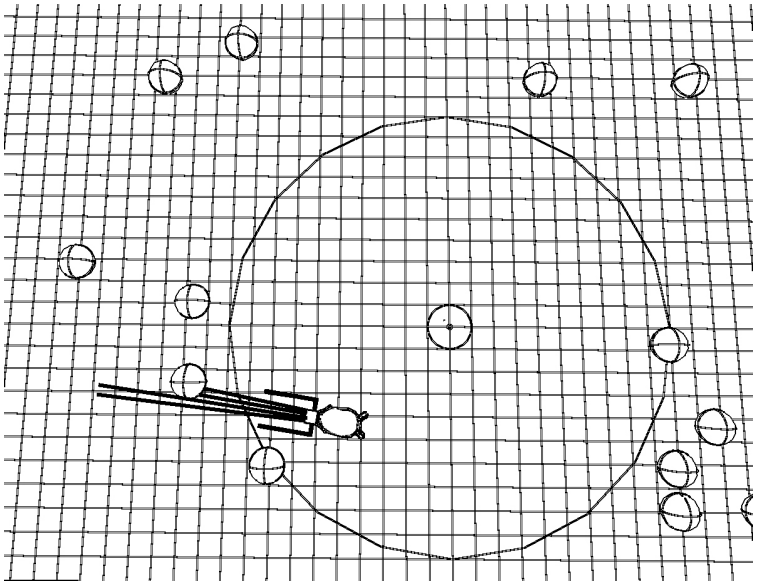


Figure 4.2: Top View of Robot in its Environment

The alternative form of the fitness function penalizes the robot for the excessive use of energy. Energy expenditure of the robot is measured indirectly by considering the energy of the entire system. At each time step the kinetic and potential energy of the system is calculated for all objects. The total energy of the system at the prior time-step is saved, so that we can observe any increases in the total energy of the system (E_c in Equation 4.1 below). The total energy of the system can decline due to energy dissipation occurring in non-elastic collisions, or in work done against friction. Since the only source of energy in the system is that provided by the application of forces to the robot, all positive increases in the energy of the system are attributed to energy expenditure by the robot. Since there are no springs in the robot, we do not need to consider the transference of kinetic energy to that of potential energy in joints.

This approach yields only an approximate estimate of energy used by the simulated robot, as it neglects work done by the robot in decelerating bodies in the system. However, it is likely to be sufficient for making estimates of relative energy expenditure between simulated robots when averaged over 10 sub-trials. We use this value (E_c) to calculate an *Energy Efficiency Factor* ρ, that tends to

one for low energy usage and zero for high energy usage in accordance with the function;

$$\rho = \frac{e^{-(E_c - k_1)/k_2}}{(1 + e^{-(E_c - k_1)/k_2})} \tag{4.1}$$

Where:

ρ is the Energy Efficiency Factor.

E_c is the cumulative positive changes in kinetic plus potential energy of the system.

k_1 is a bias term chosen to be 600.

k_2 a scale term that is chosen to be 150.

The Energy Efficiency Factor is multiplied by the fitness term used to calculate fitness in systems that are not energy constrained. For example, in the case where the robot acquires 10 spheres in 10 trials and attains an unadjusted fitness value of approximately 0.9 (due to the early termination condition), this value is multiplied by the Energy Efficiency Factor ρ. If the Energy Efficiency Factor evaluates to 0.5, then the fitness of the robot is taken to be 0.45. The values of k_1 and k_2 were chosen heuristically so that a wide range of sensor-motor "behaviours" were not excessively penalized, but those that appeared to rely largely on exploiting the maximum motor output capacities of the robot were heavily penalized. The values were chosen such that in an initial population most individuals would typically exhibit an Energy Efficiency factor of between 1.0 and 0.5. Robots that made continuous use of the maximum linear and rotational forces available to the robot would typically exhibit an Energy Efficiency factor of less than 0.1.

The genetic algorithm employed is the same as that described in chapter 3. The neural network formulation corresponds to that of the conventional GasNet described in the same chapter.

4.4 Experimental Results: Initial Results

Initially three scenarios were tested;

1. No Energy Penalty, Ray Sensors.
2. Energy Penalty, Ray Sensors.
3. No Energy Penalty, No Ray Sensors.

The graphs shown in Figure 4.3 and Figure 4.4 show the performance of the robot with and without the fitness function which penalizes excessive energy use (scenarios 2 and 1 respectively). They display the average results of 5 trial runs over 80 generations. The maximum theoretical fitness is 1.0, although this is practically limited to nearer 0.9 due to the early exit condition utilized. In both cases average and peak absolute fitness values were very similar. However, robots that were not penalized for the excessive use of energy expended nearly twice the energy of robots that were subject to penalization in the case of excessive energy use. This demonstrates that we were able to achieve results that are considerably more energy efficient by incorporating an energy penalty without sacrificing absolute performance. It is suggested by these results that consideration of energy expenditures in evolving gaits for legged robots may well benefit from the use of fitness function that takes into account energy expenditure.

Robots that were not energy constrained did not appear to be making use of sensory data in achieving their high level of fitness, but appeared to rely upon energy intensive motor activity. The fittest individuals achieved a high fitness score by rotating rapidly whilst spiralling outwards from the origin. We may have expected robots to exhibit angular deceleration and an increase in forward motion towards detected spheres, if they were making use of their sensory data.

In order to test this hypothesis, we carried out further tests where sensors were disabled (in robots that were not subject to energy penalties), listed above as scenario 3. Over 60 generations, an average fitness level of 0.849 (versus 0.842) and an efficiency (fitness over energy expended) of 2.01 (versus 1.94) were achieved. Qualitatively, the behaviours exhibited by the fittest individuals with sensors disabled were indistinguishable from those with sensors enabled. The

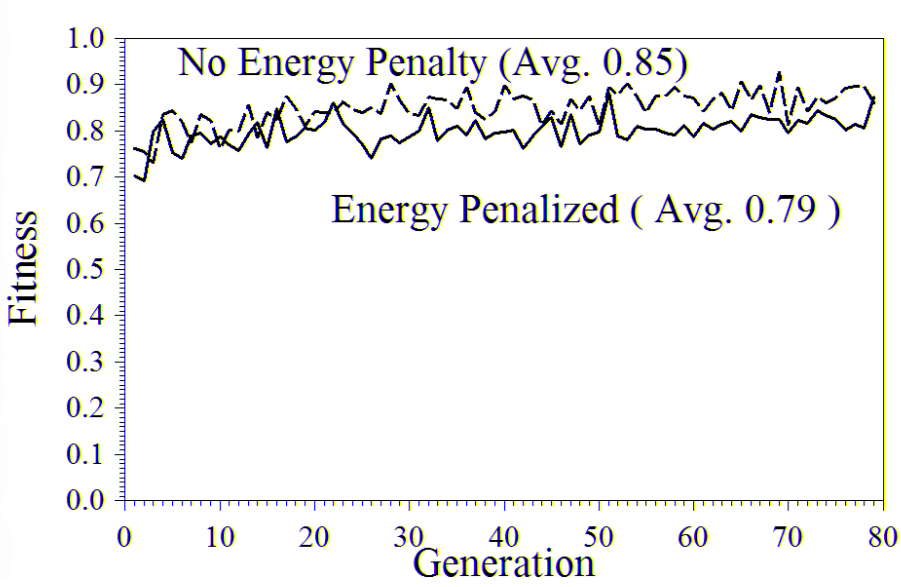


Figure 4.3: Absolute Fitness

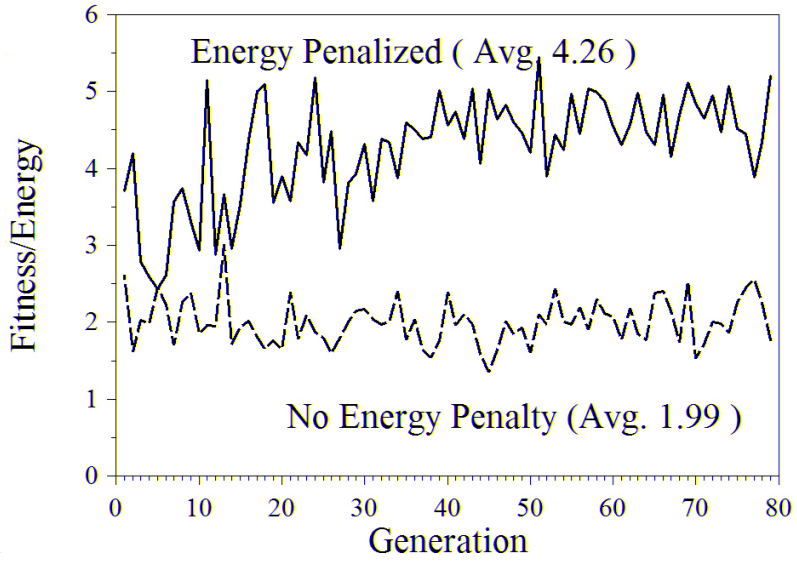


Figure 4.4: Fitness Per Energy Expended

qualitative and numerical similarity of results in scenarios 1 and 3 suggest that the robots which were not subject to energy penalties made little use of their sensory data.

4.5 Experimental Results: Secondary Results

Initial results suggested that there were two possible routes by which high levels of fitness could be attained; either with energy intensive motor activity or through active use of sensor data. In order to explore this hypothesis further a second set of scenarios were tested. This time the results were calculated over 35 generations with 5 trials for each scenario. It was hypothesized that factors affecting the relative importance to fitness of exploiting sensor data might include the utility of the sensory data, and the sparseness of prey in the environment (in this case prey being the spheres themselves). In order to investigate this, the value of the sensor data was improved, by inhibiting rotational movement on the detection of a sphere. It was assumed that an optimal low-energy strategy would involve a two stage process, where the robot would initially orientate in the direction of a detected sphere and then engage in linear motion towards the sphere. In effect, by coupling the raycast sensors to lateral inhibition, we were attempting to increase the 'value' of sensor inputs. Additionally the number of spheres was reduced from 24 to 18. Under these experimental conditions the following scenarios were tested and results obtained:

1. No Energy Penalty, No Sensor Data; Average Fitness 0.77, Fitness/Energy 1.33.

2. Energy Penalty, No Sensor Data; Average Fitness 0.69, Fitness/Energy 2.36.

3. Energy Penalty, Sensor Data; Average Fitness 0.89, Fitness/Energy 6.03.

These results help to corroborate the idea that at least under these experimental circumstances energy intensive motor activity (in scenario 1 above) can more than compensate for a lack of sensory data (scenario 2), since we are able

to achieve a higher level of fitness without sensor data, when there is no penalty for excessive energy use. In scenario 3 however, we see that the improved value of the sensor data ensures that in absolute terms, even with energy intensive motor activity, the same absolute levels of fitness cannot be achieved. This preliminary result suggests that incorporating energy penalties into fitness measures may be a useful strategy in encouraging the use of sensors in evolved behaviours, which in turn may help with the evolution of more complex sensorimotor capabilities.

4.6 Future Work

An obvious extension of this work is to incorporate measures of energy efficiency into fitness functions used for the evolution of locomotion in legged robots. Simulations involving the evolution of locomotion in legged robots typically employ fitness functions that assign the highest fitness to solutions which enable the robot to travel the furthest distance in a fixed amount of time. Clearly this biases results to gaits that make use of the maximum available energy. Experiments have shown that horses select the most energy efficient gait appropriate to a given speed (Hoyt and Taylor [1981]). Similar results have also been found for humans, and kangaroos Alexander [2003]). Incorporating energy efficiency into fitness functions should allow us to evolve gaits that are optimized for energy efficiency.

Another challenge is in evolving neural circuitry that is capable of autonomously switching between gaits to minimize energy expenditure over a range of speeds. There are two parts to this problem. This first part relates to a mechanism of switching, so that we can modify neural networks dynamically to produce the required range of gaits. One possible solution is to continue with neural networks that incorporate models of neuromodulation. Neuromodulators such as dopamine, octopamine and serotonin are known to have the capacity to chemically 're-wire' motor circuits (Kiehn and Katz [1999]). A modification of the GasNet model described in this paper may be of value in developing such switching circuits.

The second part of the problem relates to a requirement to provide the robot with a simple metabolism, or at least a method of providing some input into the

artificial neural network that reflects energy expenditure. By incorporating the "proprioception" of energy expenditure within the model, we are providing an evolutionary pathway by which energy conservative gaits can be evolved. Neural sensors that detect excessive energy use could be used as switches to trigger alternative motor gaits.

4.7 Discussion

We can perhaps imagine a continuum of experimental scenarios with variations in the scarcity of prey, and the acuity and utility of sensors. In an environment where prey is abundant and there are readily accessible supplies of energy, then the value of energy intensive motor activity in improving fitness may well diminish the importance of sensors. Conversely in an environment with sparse prey, the relative value of sensors in improving fitness is increased. This has repercussions if our primarily goal is to seek to evolve agents that make full use of sensory data in the solution of a task.

The imposition of an energy constraint changes the fitness landscape, such that robots which *do* make use of sensory data have an evolutionary advantage. A failure to impose energy penalties in evolutionary simulations reduces selection pressure on evolved entities, such that they may not necessarily take full advantage of the sensory data that is accessible to them, but may discover energy intensive motor solutions to achieve the same effective fitness. This is a strong indication that imposing energy penalties may well play a useful role in helping us to achieve 'intelligent' agent activity.

Chapter 5

Experiment: A Hand-Coded Finite State Machine

5.1 Introduction

This chapter relates to an experiment to hand-code a solution to the ball collection experiment described in the previous chapter. Evolved neural network control solutions did not result in behaviour that we would typically describe as 'intelligent'. By hand-coding a solution (described in this chapter), we are able to identify architectural features are required to solve this class of problems. By suggesting this a working hypothesis for some attributes of brain anatomy, we attempt to identify homologous brain regions.

The motivation for this chapter is simply explained. The failure of GasNets to present with a convincing solution to this class of problems, implies that we are lacking circuit models that better suited to this problem class. Since the over-riding goal of the work described in this thesis is; to identify complementary circuits suitable for coevolution; to seek to devise circuits that are inspired by biological systems, identification of the brain region that is implicated in the solution of such problems is of particular value. Through concentrating on this brain region, we are then able to gain some inspiration for models that might allow us to abstract the necessary characteristics, in a form that is ultimately tractable to evolutionary search.

5.2 Experimental Task

In a similar fashion to that described in Chapter 3, the experiment involved the acquisition of balls by a robot in a physically simulated environment. In terms of the simulation, the physical structure of the robot was simplified as shown in figure 5.1.

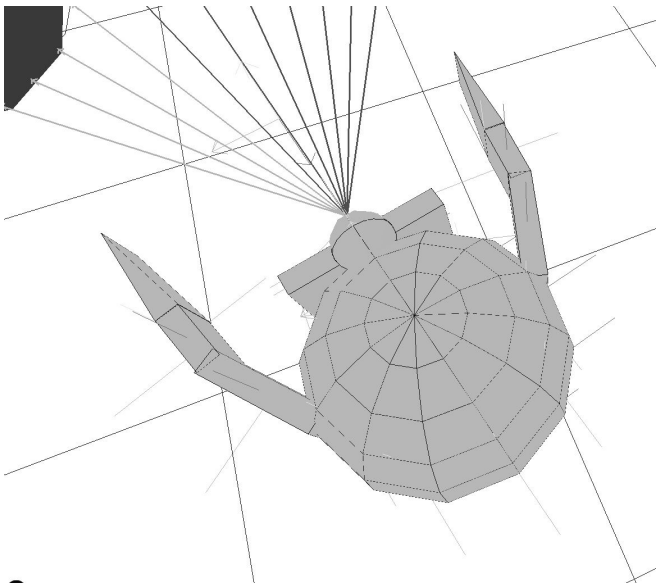


Figure 5.1: Perspective View of robot and Sensor Rays

5.3 Code Implementation

The schematic displayed in figure 5.2 shows details of the structure of the hand-coded solution. No attempt was made to ensure that the implementation was biologically realistic or plausible, only that its behaviour should appear relatively intelligent and energetically efficient. The implemented solution was intended to reflect the conception of a nominally optimal solution. The robot should wander when there is no food within sensor range. Once food is detected, the robot should seek to minimize its translational velocity and orient itself with the food so that the food lies directly ahead. Once it is appropriately oriented it approaches the

food. When the food is discerned as being within the extents of the robot's claws, it seeks to remaining stationary until the food is "eaten".

Sensor data is first pre-processed, so that information from the array of sensor rays is converted into a form that is readily of use. We have a low-level sensor buffer that stores information which relates to whether food is detected to the left, to the right or directly in front of the robot. Additionally we have a high-level sensor buffer that stores data which relates to whether or not food (i.e. a ball) has been detected, and separately, whether or not food is within the extents of the robot's claws. The sensor data is effectively pre-processed such that the dimensionality of the data is reduced, and salient signals extracted.

At the output stage, we have a motor buffer that accepts control signals from the robot. These control signals relate to the ability to apply a linear or rotational impulse. These actions are effectively independent and are not mutually exclusive. We can apply a rotational impulse at the same time that we apply a linear impulse. Each motor activity has its own sub-program. They are two key commands that relate to stopping linear motion, and stopping radial rotation. These ensure that the motor sub-programs seek to attain a specific control variable value, that of zero translational velocity and zero rotational velocity respectively.

The contents of the high-level sensor buffer are used to determine a behaviour mode. These are hard-coded as;

- Wander
- Approach
- Orient
- Eat

These behaviour modes correspond to a single state variable that determines which control sub-programs are executed, and what interrupts are relevant to changing state. Once we are in one behavioural mode, we do not escape until prescribed signals have been detected. These signals may relate either to a successful completion of a task or the failure of a task. To give a concrete example, when we are in "Eat" Mode, if the contents of the high-level sensor buffer indicate

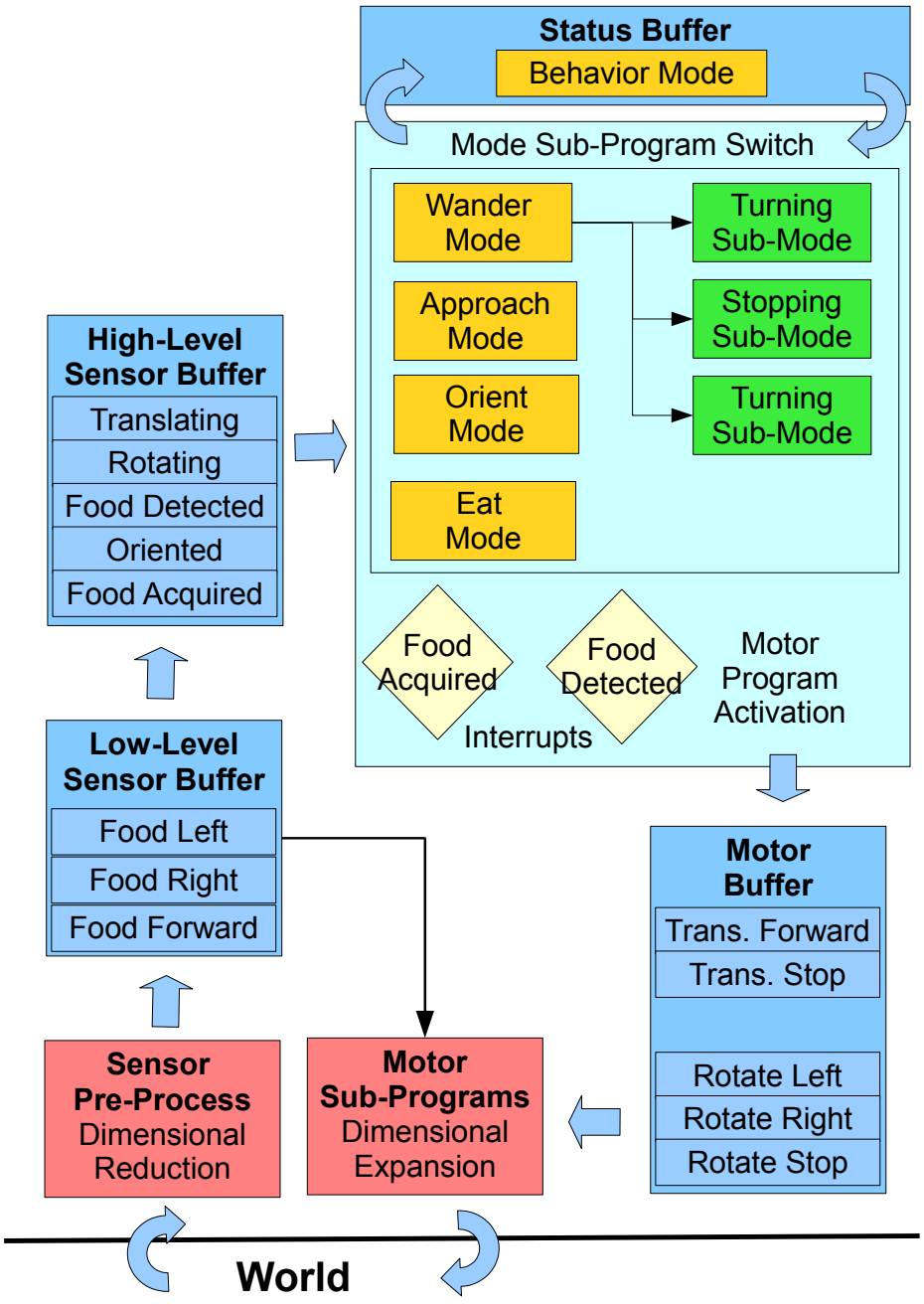


Figure 5.2: FSM robot Schematic

that we are no longer within eating range of the ball, we examine the high-level sensor buffer to see if we remain appropriately oriented. If so, we switch into "Approach" mode, otherwise we switch into "Orient" mode. If for some reason we are out of range of the ball altogether, then we switch into "Wander" mode. The Wander mode itself has sub-modes associated with it, corresponding to sub-control programs. Note that any of the sub-modes can be interrupted by the detection of food.

The code implementation efficiency of decomposing the task into these sub-modes arises from the fact that, for each sub-mode we only have to consider a sub-set of all sensor data that is relevant, and execute only a sub-set of all possible motor control programs. Initially, the converse approach was taken where we examined the entirety of the sensor data to determine the appropriate motor response. This proved to be less efficient, since it does not in itself easily support a chain of events. Intrinsically, if we are in "Orient" mode, and we lose contact with a ball, then the only alternative is to enter "Wander" mode. We are able to bring domain specific knowledge, which enables us to minimize the number of sensor states that we need to consider, before entering the next behavioural sub-program.

Another area where we are using domain specific knowledge is in the decision to create low-level and high-level sensor buffers. The behavioural control program does not need to know if there is food on the left or the right. It only needs to know if food has been detected, and how close it is, to determine whether or not to change behavioural modes. The motor control programs, however, make direct use of this low-level data. If we are in "Orient" mode, the motor control program will determine in which direction it should turn, by looking at this low level data.

Part of the code implementation is shown in the listing 5.1.

Listing 5.1: The "Approach" Switch Clause

```
case eCPNRB_Approach:{

        // Confirmatory State.
        // (a)  If food has been detected forward and
        //              we are currently stationary or moving,
        //              and the food has not been acquired.
        if(             (pNStatus[eSTATUS_FLAGS] & eSDV_Food_Detected )
                &&      (pNStatus[eSTATUS_FLAGS] & eSDV_Oriented )
                &&      !(pNStatus[eSTATUS_FLAGS] & eSDV_Food_Acquired )
                ){
                pNMotors[ eCMD_LIN ] = eMDCL_Translate_Forw;
                }
        else if(        (pNStatus[eSTATUS_FLAGS] & eSDV_Food_Detected)
                &&      !(pNStatus[eSTATUS_FLAGS] & eSDV_Oriented)
                &&      !(pNStatus[eSTATUS_FLAGS] & eSDV_Food_Acquired)
                ){
                // BEHAVIOR CHANGE to eCPNRB_Orienting
                // (b)  We are not longer aligned, but we
                // can still detect the food. In this
                // case we need to re-align.
                pNMotors[ eCMD_LIN ]            = eMDCL_Translate_Forw;
                pNStatus[ eSTATUS_BEHAVE ]      = eCPNRB_Orienting;
                }
        else if((pNStatus[eSTATUS_FLAGS] & eSDV_Food_Acquired )
                ){
                // BEHAVIOR CHANGE to eCPNRB_Eating.
                // Start eating if the food is acquired.
                pNMotors[ eCMD_LIN ]            = eMDCL_Translate_Stop;
                pNMotors[ eCMD_ROT ]            = eMDCR_Rotate_Stop;
                pNStatus[ eSTATUS_BEHAVE ]      = eCPNRB_Eating;
                }
        else if( !(pNStatus[eSTATUS_FLAGS] & eSDV_Food_Detected) ){
                // BEHAVIOR CHANGE to eCPNRB_Wander.
                // Start wandering if the food cannot be detected.
                pNMotors[ eCMD_LIN ]            = eMDCL_Translate_Stop;
                pNMotors[ eCMD_ROT ]            = eMDCR_Rotate_Stop;
                pNStatus[ eSTATUS_BEHAVE ]      = eCPNRB_Wander;
                pNStatus[ eSTATUS_SUBMDE ]      = eCPNRS_NULL;
                }

        }break;//Behavior eCPNRB_Approach
```

Note that there are two key buffers; the **Status Buffer**, pointed to by pNStatus, from which data is extracted, and the **Motor Buffer**, pointed to by pNMotors, into which data is written. Although there is also a **Sensor Buffer**, it is not needed in this switching clause, since the data in the Sensor Buffer has

been pre-processed, with the results placed in the Status Buffer. This is an example of dimensionality reduction. The clause itself relates to the "Approach" behaviour mode. As we enter this clause, we check certain Status Buffer flags. Depending upon the contents of these flags, we then either issue Motor Commands (by writing to the Motor Buffer), or change the behaviour mode by writing to the Status Buffer.

The Motor Commands themselves, correspond to high-level commands, for example, when we update the Motor Buffer slot;

```
eCMD_LIN
```

with the value;

```
eMDCL_Translate_Stop
```

we are ensuring that that the motor control program adjusts the linear velocity of the robot to zero (through the appropriate application of linear impulses). In parallel with this, we can also update to the Motor Buffer slot;

```
eCMD_ROT
```

with the value;

```
eMDCR_Rotate_Stop
```

which will ensure that a separate control program reduces the rotational velocity of the robot to zero. There is no need to conflate the command clause with either low-level sensor data or low-level command data.

5.4 Results

Although no numerical comparison was made between the experimental results achieved in Chapter 3, the qualitative results conform far more to what we would might regard as intelligent behaviour.

Screenshots of the robot engaged in three of the behaviour modes are shown in the following figures; Wander 5.3, Approach 5.4 and Eat 5.5.

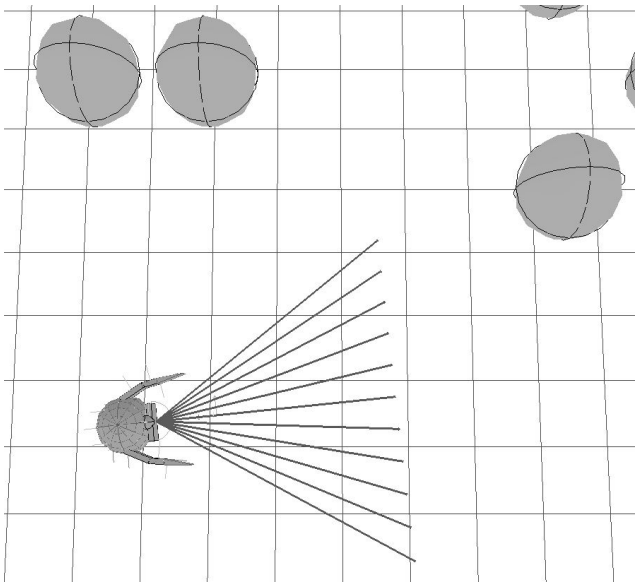


Figure 5.3: FSM robot in “Wandering” Mode

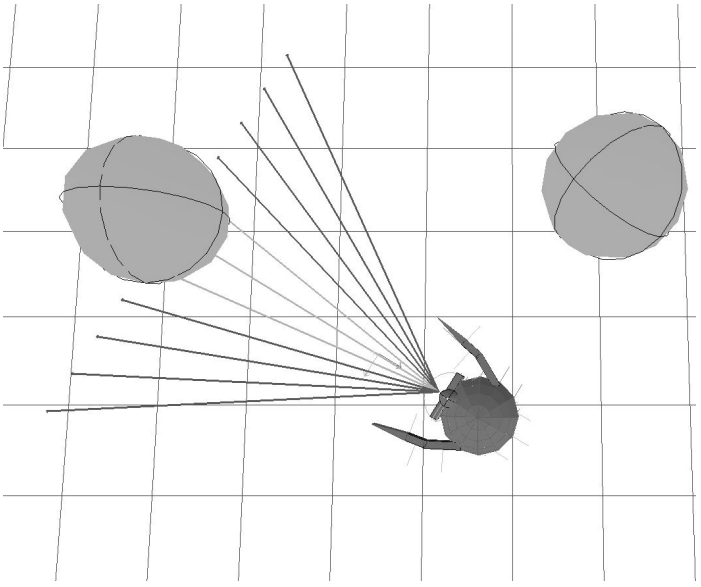


Figure 5.4: FSM robot in “Orient” Mode

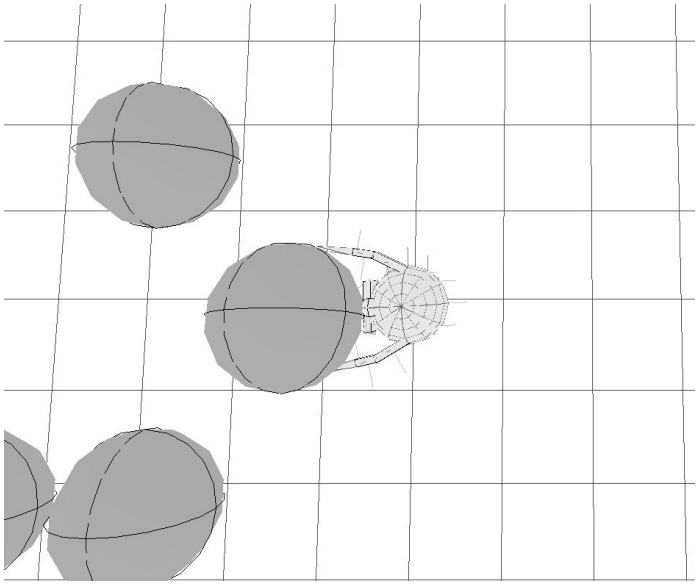


Figure 5.5: FSM robot in “Eat” Mode

The goal of the work described in this chapter was to see if we could gain some insight into deficiencies in the evolvable neural network models that, whilst successfully applied to locomotion (in chapter 3), appear to be of less use in addressing problems that are more easily solved through a Finite State Machine (FSM) like architecture. In the solution of this problem, we are able to discern a number of architectural features that may be of value;

- Behaviour is divided into behaviour-modes, where only sensor data is considered and appropriate motor units activated that are relevant for the mode.
- Behaviour-mode changes take place due to mode specific interrupt conditions.
- Sensor data is dimensionally reduced prior to processing by the unit that governs mode switching.
- Some low-level sensor data is fed directly to motor programs without being considered by the behaviour mode switch.

- Motor sub-programs running in parallel are activated by signals from the behaviour mode switch.
- Low-level motor-programs typically operate by seeking to maintain control variables at certain values. The relevant control variable at any time is set by the higher-level motor control,

5.5 Discussion

5.5.1 A Modified Ashby Perspective

These represent features of a system that is essentially mechanistic. It seems reasonable to consider how we might map the architectural features of this program to some kind of analogue system. Despite the fact that we have constructed a mechanistic solution to the problem, there may be something about the structure of information through the system that would be invariant with an analogue interpretation. Figure 5.6 shows how might interpret the program architecture from a modified Ashby perspective (see section 2.5.2). Note that in this diagram only some of the motor programs are illustrated.

We are able to identify certain aspects of the program that are immediately recognizable as corresponding to Ashby's notion of an *ultrastable system* (see figure 2.10). In the diagram, these are motor programs that relate to orientation and reducing rotational and linear velocities to zero. The diagram in its current form illustrates the activation of two motor programs (orientation and stopping). This corresponds to the orientation 'behaviour' mode. Under these circumstances the robot has already detected food, and has some measure of the difference between the robots current orientation direction and that of the food itself. This measure represents an *essential variable* to use Ashby's terminology. For the purposes of ensuring that the appropriate orientation is achieved, the the motor program responsible for orientation, adjusts the *parameters* (S), so that the *essential variable* for this *ultrastable* unit tends to zero. In order that this is achieved efficiently the motor program for reducing the linear velocity of the robot to zero is also active. In this case the *essential variable* corresponds to the linear velocity, in which case

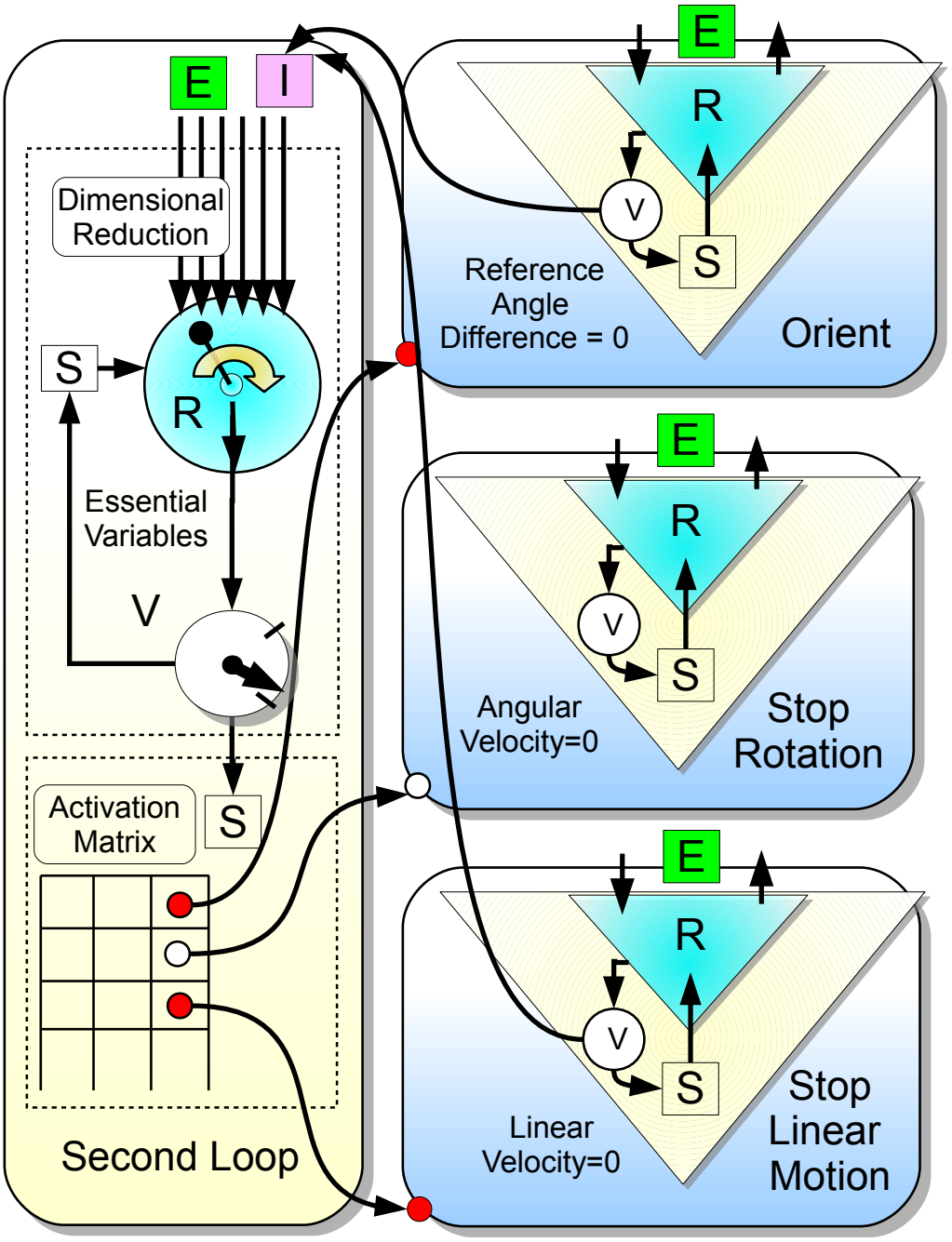


Figure 5.6: **The architecture from a modified Ashby perspective**: (E) represents the *environment*, (R) represents the *reacting part*, (S) represents the *parameters*, (V) represents the maintenance of *essential variables*, (I) represents part of a feedback loop for internal information. Note that the **Stop Rotation** motor program is deactivated in this example, whilst 'orient' and 'stop linear motion' sub-programs are activated.

parameters (S) of this *ultrastable* system are adjusted so that this value tends to zero.

The next significant feature that we are able to distinguish is that of the *second loop*. The *second loop* in Ashby's model corresponds to the secondary feedback loop associated with changing the *parameters* of the *reactive part* (R). Here we differ somewhat from the standard model of an *ultrastable* system in that the *parameters* of the *second loop* correspond to motor program activation switches. Rather than a single *reactive part* (R), we have multiple motor programs operating in parallel. It is not conceptually difficult however to imagine these motor programs as comprising a single *reactive part* (R), at least from the perspective of the *second loop*.

Note that in the *second loop*, the *essential variables* that we are maintained, are those that correspond effectively to the behaviour mode states; Wander, Approach, Orient, Eat. This is where we deviate again from the conceptual model employed by Ashby. Ashby's conception of *essential variables*, were those state variables that were maintained for the purposes of supporting survival. Ashby's notion assumes that these variables are to be maintained for the duration of a given task (in the case of the homeostat for example). Ashby's model has no explicit notion of state sequences as the possible building blocks of adaptive behaviour. Yet such sequences are possible through the persistent perturbation of essential variables outside of their predetermined values. Implicit in Asby's perspective is the feeling that the goal of the system is to maintain these stable states. Without changing the architecture of an *ultrastable system* it is also possible to conceive of an alternative interpretation. Such an interpretation might regard dynamic instabilities in the system as 'good', since it is these instabilities that might in themselves elicit state sequences that correspond to adaptive behaviour. Under such circumstances the focusses switches from the preservation of essential variables, to the appropriate support for state interrupts, that allow for the construction of action sequences.

This requires an elaboration of Ashby's notion of an *ultrastable system*. In our modified perspective, the *essential variables* correspond to a pattern that is to be associated with a given set of interrupts. All this takes place within in the *second loop*. Ashby's notion of *essential variables* that are to be maintained, is replaced

by the notion of *essential state*, that is preserved until the necessary interrupts destabilize this state. In a system capable of supporting adaptive behaviour it should be the case that there exists a mechanism (if it is not genetically predetermined) whereby the subsequent state (and its associated interrupts), *after* destabilization, is that which supports the organisms goals of survival.

Note that we have not arrived at such a conceptual model arbitrarily. What we have done is to describe the conceptual model corresponding to the hand-coded solution that we have implemented. In our case, the *essential state* transitions are deterministic and predetermined. However, there is no reason why the same conceptual model might not be implemented in some fashion whereby the state transitions are able to be learned.

5.5.2 The Brain - A First Approximation

We do not seek to create solutions that might appear attractive and expedient from a computational or mechanistic perspective, since we are already aware of the limitations of such an approach (see sections 2.3.4.2 and 2.3.4.4). Rather, our intent is to gain an insight into biological process, so that we might determine an appropriate abstraction of them. With this in mind, we shall take the naive perspective of considering, to a first approximation, how the architecture illustrated in figure 5.6 and described in the previous section, maps to the brain. The results of such a thought experiment are illustrated in 5.7.

A justification for the diagram illustrated in figure 5.7 requires an understanding of the role of the brain physiology particularly as it relates to the basal ganglia and thalamus. We carry out a review of the basal ganglia and its relationship with the thalamus and cerebral cortex in the next chapter (chapter 6). The results depicted are based primarily on the connection patterns and functional characteristics of the differing brain regions. Note, that we are not making any definitive claims regarding the functional role of brain anatomy. We are simply stating that *if* our hand-coded architecture captures some of intrinsic features required to support behaviour sequences, and *if* our functional description captures some of these features (based on a modified Ashby perspective), *then* figure 5.7 represents a reasonable mapping of these functional features to brain anatomy.

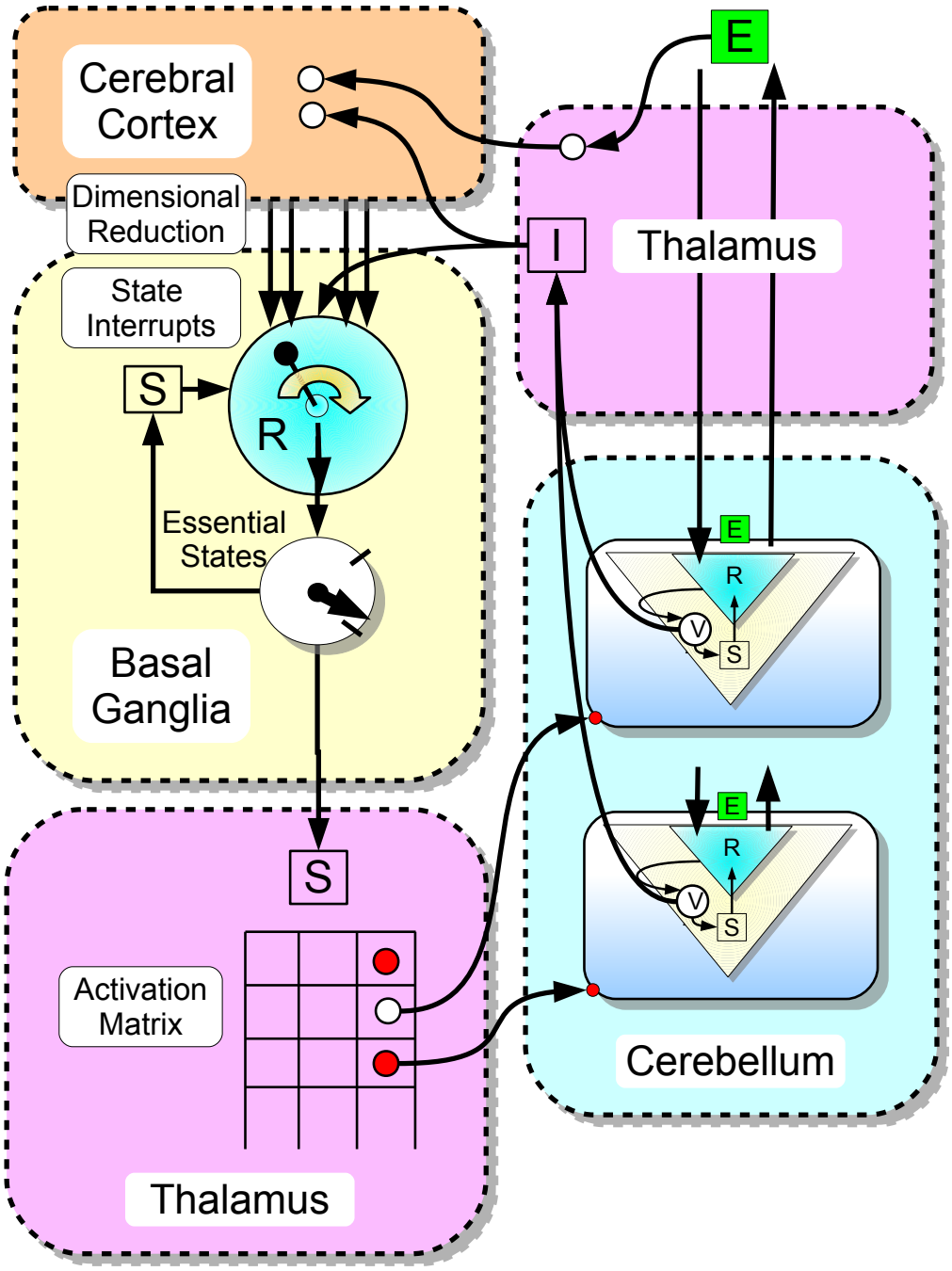


Figure 5.7: **The Brain from a modified Ashby perspective**: (E) represents the *environment*, (R) represents the *reacting part*, (S) represents the *parameters*, (V) represents the maintenance of *essential variables*, (I) represents part of a feedback loop for internal information.

5.6 Conclusion

What should be readily apparent when considering brain anatomy is that there is a macro-level structure, and micro-level structure. Different brain regions comprise neurons that have functional characteristics that are very typical of their local region, but atypical of neurons found in other regions. Some regions are relatively homogeneous in their neuron types, others are highly diverse.

Despite this, most evolutionary robotics experiments typically only explore one particular circuit form, with neurons that are relatively homogeneous in functional characteristics. Very little consideration is given to the prospect of creating variation at the macro-level, supporting functional diversity at in differing brain regions. It is therefore perhaps not surprising that evolving complex behaviour can be difficult. Given the suitability of GasNets in support of locomotion, it would be rather surprising if the same circuit formulation was universally applicable to all problem domains. There is a reason why different brain macro-structures have different neuronal composition, and that is to support a range of specialized processes.

In a hand-crafted solution to a finite-state-machine like problem, we arrive at an architecture that maps quite well to macro-structures in the brain (cerebellum, thalamus, cerebral cortex and the basal ganglia), and conceptually bears some similarity to a modified version of an ultrastable system. As we outlined in section 2.5, the central challenge in evolutionary robotics is to extend the boundaries of complex adaptive behaviour. Our preferred solution to this problem is to identify (or create) circuits that might provide us with complementary functional characteristics.

Taking a very simplistic view, we know of the existence of circuit forms that in some sense resemble characteristics of the cerebellum (e.g. supervised learning, GasNets in support of locomotion), and the cerebral cortex (e.g. unsupervised learning, self-organizing feature maps). The thalamus is known primarily for its routing functionality, analogous in some sense to a complex matrix communications channel switch. This leaves us with the basal ganglia. Whilst it is deemed to be functionally implicated in reinforcement learning, we lack connectionist models to the same extent that they exist in modelling other brain regions.

In chapter 4, we failed to evolve GasNets to solve this task in a fashion that we might regard as 'intelligent', analysis of our hand-crafted solution and potentially homologous structures in the brain, implies that circuits functionally analogous to those of the basal ganglia might help us to solve this problem.

Our premise is that adaptive behaviour is best supported by circuits that exhibit complementary functionality. We already have a solution to locomotion in the form of GasNets. Circuit models for the cerebellum and cerebral cortex already exist. Some model of the basal ganglia tractable to evolutionary search, might provide us with the last remaining building block. If we are able to coevolve systems that exhibit both macro structure *and* functional specialization at the micro-level, we might be able to significantly increase the level of complex adaptive behaviour that we are able to achieve through evolutionary robotics methods.

Chapter 6

The Basal Ganglia and Dopamine Signalling

6.1 Overview

Biological studies relating to dopamine (Schultz [1998]), and machine learning models of artificial intelligence (AI) which exhibit reinforcement learning (Sutton and Barto [1998]), represent a very powerful synthesis of ideas in helping us to understand behavioural adaptation in response to rewards. Specifically, the phasic activity of dopamine neurons shows a strong correspondence to the Temporal Domain error signal, associated with reward prediction errors in reinforcement learning models (see Vitay et al. [2009] for a review or biological models of reinforcement learning). More recent studies paint a more complex role of dopamine in the brain (see more details later in this chapter), but do not fundamentally undermine the validity of the current orthodoxy. Rather they offer more detail, new perspectives, and highlight ancillary roles for dopamine as a signalling agent. In Chapter 6 we compare our conceptual model with these reinforcement learning formalisms, but we also suggest Karl Friston's Free Energy (*prediction error*) Model as a possible alternative formalism (Friston [2010]).

The fundamental goal of this research is to devise reward-anticipatory neural network models that may be tractable to evolutionary exploration for use in supporting more complex robot behaviour. Unfortunately existing computational

models, either biologically plausible models of the dopamine system or abstract models based on reinforcement learning, are typically monolithic and complex. Solving this problem necessitates a return to biological systems, with a view to understanding the underlying principles which may have given rise to the reward anticipation circuits referred to in the title of this thesis. This requires a re-consideration of some fundamental issues (such as the role of energy efficiency in signalling regimes) and, through consideration of current reward systems, involves seeking to turn back the evolutionary clock, with a view to understanding how existing reward-anticipatory circuitry might have emerged. Whilst such an approach might seem to require a consideration of primitive organisms, the extensive availability of literature on the role of dopamine in the mammal brain, and the contrary signalling uses of dopamine in more primitive organisms (e.g. dopamine signalling is associated with aversive reinforcement in drosophila melanogaster, the common household fly, Waddell [2010]), make such an approach less attractive. However since inspiration for the devised models is drawn from studies of the human and monkey brain, we are able to arrive at some ideas that relate to the functional significance of certain brain areas in the basal ganglia.

Although there are four significant dopamine systems recognized in the brain (Moore and Bloom [1979]), we focus on that which corresponds to the projection of dopaminergic neurons from the substantia nigra pars reticula to the striatum of the basal ganglia, as the most prominent of these. This system is termed the nigrostriatal pathway. There are three other systems that we do not address directly. These comprise; the tuberoinfundibular (projections from the arcuate nucleus to the median eminence of the hypothalamus), mesolimbic (projections from the ventral tegmental area to the nucleus accumbens and amygdala) and mesocortical (projections from the venreal tegmental area to the prefrontal cortex) systems.

There is some support to the idea that the basal ganglia is an evolutionary solution to the *action selection* problem amongst vertebrates (Redgrave et al. [1999b] and Stephenson-Jones et al. [2011]). We should be somewhat careful here as to how we delineate the scope of this research. The term action selection is extremely broad, in that it may includes higher level cognitive function related complex behaviour.

The focus of the work described in this thesis is more akin to that encompassed by the hand-coded model described in chapter 5. When behaviour modes are referred to, we were essentially referring to distinct connectivity patterns of selective sensor consideration and relevant motor program activation, which result in functionality distinct activities. It is only in this limited sense that the term "behaviour" or "behavioural modes" is employed within the context of this thesis.

In this chapter we highlight key areas of research that relate to the basal ganglia and dopamine, and make some reference to their relationship with the work carried out in this thesis. In order to conceptualize these areas more easily, we break them down into three sections;

1. A "top-down" perspective that sees the basal ganglia as a system (see chapter 7).

2. A "bottom-up" perspective focusses on the significance of dopamine signalling and its role in circuit adaptation to rewards (see chapter 8).

3. Computational models that represent an integration of these perspectives.

6.2 The basal ganglia perspective

Traditionally the basal ganglia has been associated with motor functions, in areas such as motor coordination and learning (Hikosaka [1991]), partly due to its association with movement disorders such as Parkinsons disease (see Wichmann and Dostrovsky [2011] for a recent review). The current orthodoxy is that the basal ganglia is functionally significant in the task of sequence learning, in contrast with the Cerebellum which is more closely associated with low-level motor control and continuous motor adaptation (Doyon et al. [2009]), although their functionality is closely integrated. More recently however, its role in non-motor functions is also widely recognized. Figure 6.1 shows a morphological illustration of the basal ganglia and its connections (Wilson [2004]). Figure 6.2 shows an augmented version of the conventional schematic (Bar-Gad et al. [2003]). A number of key concepts that are particularly relevant to this thesis are described

below in the following subsections. We shall return to these issues in Chapter 10 when we contrast the model proposed in this thesis with our knowledge of the basal ganglia.

6.2.1 Direct, Indirect and Hyperdirect Pathways

There are a number of pathways associated with the basal ganglia system, each of which is deemed functionally significant (see figure 6.3). The two most commonly referred to constitute part of the cortico-striato-thalamic loop, and are termed the *direct* and *indirect* pathways (Albin et al. [1989] and Alexander et al. [1990]). Both of these pathways deemed to originate from the striatum. Note that the striatum takes its input from the cortex and thalamus. The *direct* pathway relates to projections from the striatum directly to the substantia nigra pars reticula (SNr). The *indirect* pathway refers to projections from the striatum to the substantia nigra pars reticuala via a route that visits the globus palidus external section (GPe) and the subthalamus nucleus (STN). Nambu (Nambu et al. [2002]) proposed a slightly different definition from that which is commonly used. In so doing, Nambu sought to draw attention to the functional significance of the subthalamic nucleus, which in addition to the striatum, also receives projections direct from the cortex. He supports the idea that the subthalamus nucleus (STN) has a role in modulating the pathway from the striatum to the substantia nigra pars reticuala via the globus palidus external section (GPe). Since the functional role of the basal has yet to be conclusively determined, understanding of the potential role of cortical modulation supported by the STN has received even less attention. However by adopting Nambu's nomenclature, we are at least allowing space for future work that provides us with a more complete picture. For this reason, in the context of this thesis, the terms direct and indirect pathways will correspond to those proposed by Nambu.

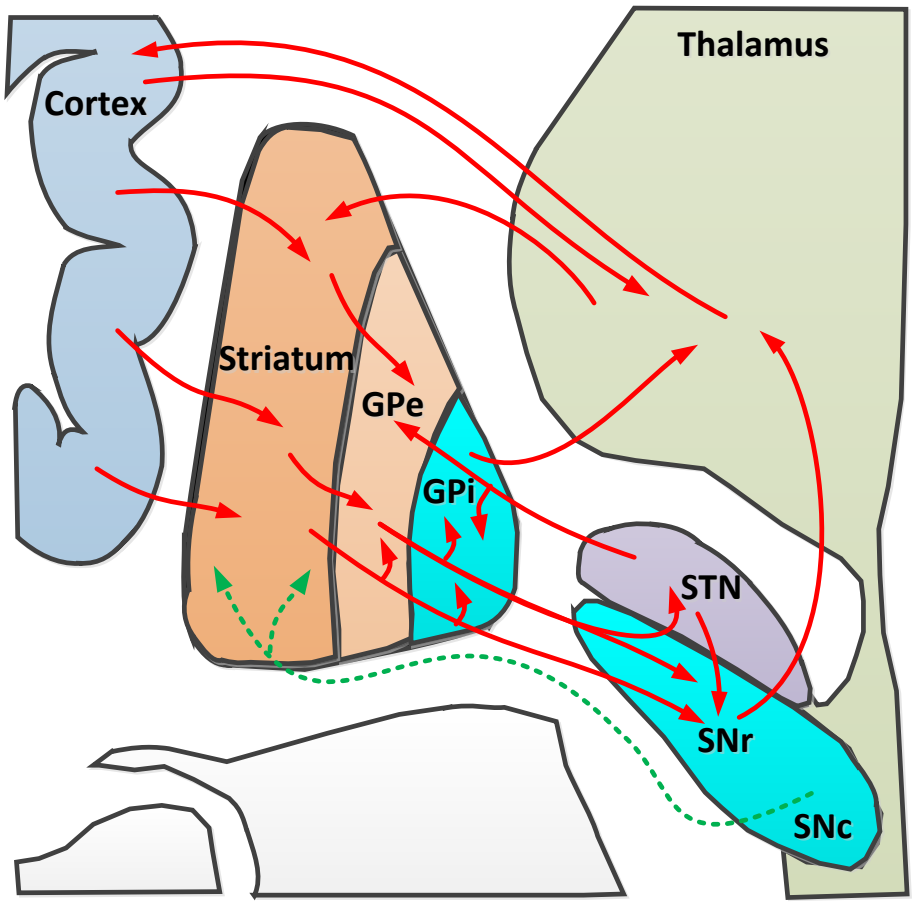


Figure 6.1: **Morphological illustration of the basal ganglia and its connections**. Note that this includes the thalamo-striatal pathway that is often excluded from schematic drawings of the basal ganglia and its connections. Note also that the GPi and SNr are physically distinct. This is also not obvious from most schematic diagrams. Abbreviations: GPe, globus palidus, external segment; GPi, globus palidus, internal segment; SNr, substantia nigra, pars reticula; SNc, substantia nigra, pars compacta; STN, subthalamic nucleus. Adapted from Wilson [2004].

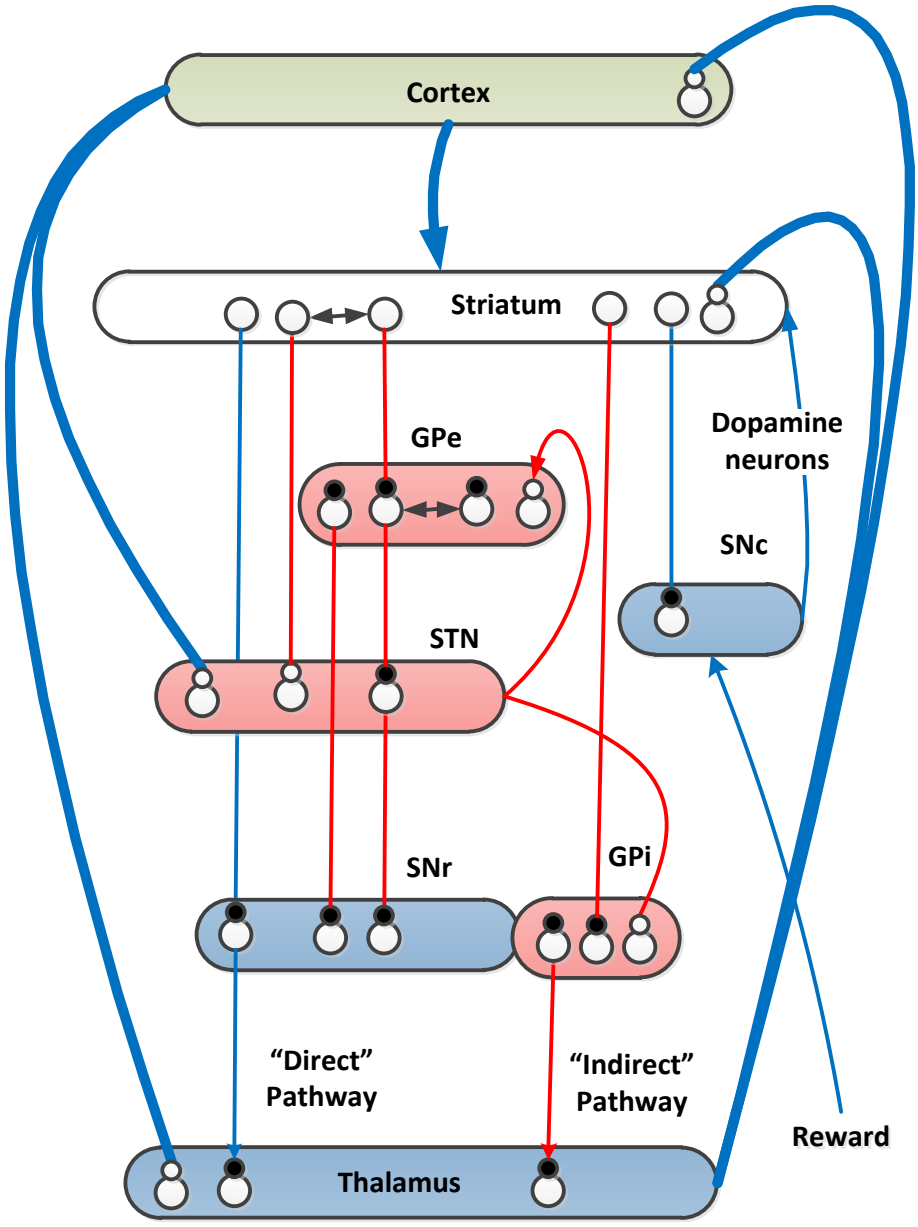


Figure 6.2: **Schematic of the basal ganglia and Associated Pathways**. Although this represents a conventional schematic it includes the thalamo-striatal pathway that is often excluded from such diagrams. Glutamatergic synapses (excitatory) are denoted by empty circles, GABAergic synapses (inhibitory) by black-filled circles . Abbreviations: GP, globus palidus, SNr, substantia nigra, pars reticula, SNc, substantia nigra, pars compacta, Sth, subthalamic nucleus.

6.2.2 Parallel Cortical Loops

The basal ganglia appears to be a striking example of the process of *exaption* (the functional reuse of prior entities through adaptation under evolution). This is most evident in the parallel loops that pass through the basal ganglia (Garret E.Alexander, Mahlon R. DeLong [1986], Garret E.Alexander, Mahlon R. DeLong [1986], Haber and Calzavara [2009]). These pathways are largely segregated, and are recognized to support distinct limbic, motor and executive (sometimes further divided into Sensor and Associative) systems, in large loops encompassing the cortex, striatum, substantia nigra, palidum and thalamus.

6.2.3 Subcortical/Nauta-Mehler Loop

A second single loop is that which links the striatum, globus palidus interior segment and the Thalamus, in what has been termed a "Subcortical Loop" (McHaffie et al. [2005]). This loop is sometimes termed the Nauta-Mehler Loop, in recognition of the seminal work carried out by Nauta and Mehler (Nauta and Mehler [1996]) identifying key pathways in the basal ganglia. Although this pathway has been recognized by a number of researchers (Kim et al. [1976], Giménez-Amaya et al. [1995], Mengual et al. [1999a] and Lanciego et al. [2004]), it receives far less attention than the Cortical Loops, and is often completely absent in schematics of the basal ganglia. This route is implicated in responding to novel events that are not explicitly reward related. This is something that we shall return to in Chapter 11.

6.2.4 Contrasting Views of the SNr and GPi

Perhaps one of the reasons for the failure to take into account the Nauta-Mehler Loop in most accounts of the basal ganglia, is that conventionally the substantia nigra pars reticula (SNr) and globus palidus interior segment (GPi) are often perceived as fulfilling the same role. They are similar in cellular structure, and developmentally originate from the same source. Most schematics (including that by Nambu displayed in figure 6.3) combine the two, despite their being physically distinct (see figure 6.1). Whilst some researchers argue for an independent view

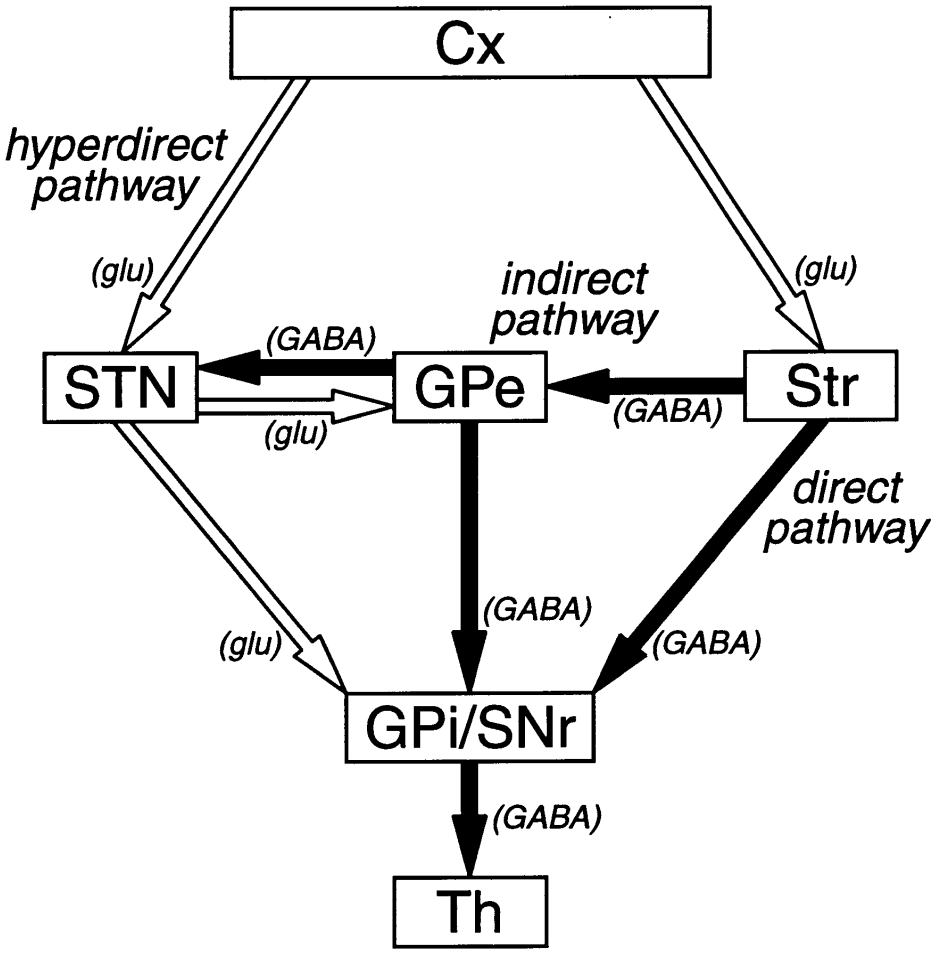


Figure 6.3: **Direct, Indirect and Hyperdirect Pathways**. A variation on the conventional "Direct" and "Indirect" Pathways is that proposed by Nambu (Nambu et al. [2002]) intended to highlight the conjectured role of the subthlamaic nucleus as a modulator of the Indirect pathway. Abbreviations: Cx, cortex; GPe, globus palidus external segment; GPi, globus palidus interior segment; SNr, substantia nigra, pars reticula; STN, subthalamic nucleus; TH, thalamus; glu, pathways that employ glutamatergic synapses (excitatory); GABA pathways that employ GABAergic synapses (inhibitory). Source Nambu et al. [2002].

of the globus palidus interior segment (GPi) and substantia nigra pars reticula (SNr) (Parent and Hazrati [1993]) this perspective can be quite strongly opposed (Hironobu Tokuno [1993]). Although this is the minority view, we take the position that they should be seen as distinct bodies. Despite a very close similarity at the cellular level, their outputs imply that they support different functionality. Again, this is an issue that we shall return to in Chapter 11.

6.3 The Dopamine Perspective

The basal ganglia perspective focusses on its role from a holistic perspective, with an emphasis on connectivity with other brain systems. As such it can be regarded as a "top-down" view. As far as dopamine is concerned, we are principally interested in the role of the dopaminergic neurons that project from the substantia nigra pars compacta (SNc) to the striatum. Such a view could be characterized as a "bottom-up" view of the basal ganglia. Clearly the two perspectives address many of the same issues. Much of our insight into the role of the basal ganglia has been gleaned from consideration from this "bottom-up" perspective.

The number of papers listed on PubMed that relate to dopamine is currently around 100,000 (Björklund and Dunnett [2007]). The classical neurotransmitters arranged in order of the number of papers currently produced each year are; dopamine, serotonin, acetylcholine, noradrenaline. Whilst the annual publication of research papers that concentrate on acetlycholine and noradrenaline is currently running at around 2,000 a year, the number is double that for dopamine (and serotonin) at 4,000.

In seeking to synthesize the results of these papers, we are helped by the fact that a significant theme running through this research is the functional role of dopamine in the brain and core ideas that were laid down in the 1990's, indicating that dopamine phasic activity in the basal ganglia corresponds to Reward Prediction Error (Schultz et al. [1998]). The emergence of ideas on reinforcement learning (Sutton and Barto [1998]) as a significant formalism in the Machine Learning community occurred at around the same time. These ideas provided the biological community with an explanatory framework for the interpretation

of their results. Conversely the Machine Learning community was able to refer to results in neuroscience which imply that processes of this kind are relevant to biological systems. The complementary nature of these two fields, and the correspondence in particular between the role of dopamine and aspects of models of reinforcement learning, has had a significant impact both in neuroscience and in the application of these findings to Artificial Intelligence.

Over the last 20 years, a considerable amount of work has been carried out that has broadly expanded these initial findings. This includes changes in our understanding of dopamine from neuro-chemical (see Björklund and Dunnett [2007] for a review), neurocomputional (see Cohen and Frank [2009] for a review) and reinforcement learning perspective (see Vitay et al. [2009] for a review), which in turn has resulted in increasingly refined ideas. A number of reviews have been written that relate to these shifting paradigms (e.g. Salamone [2007]), including one by Wolfram Schulz (Schultz [2010]).

Some of the key contributions in this area are referred to herein in order to help illustrate the complexity of this field. Some criticism of the original interpretation of results has been made on the grounds that dopamine response time is too short to coincide with prediction errors (see Redgrave et al. [1999a] and Redgrave et al. [2008]). Other researchers have proposed alternative interpretations, such as that of "Incentive Salience" (Berridge [2007]), rather than reward prediction error, as a more appropriate interpretation of dopamine phasic activity. More complex models have emerged that seek to relate dopamine to rewards, aversive behaviour and motivational control (Bromberg-Martin et al. [2010]) and "Partial Observability" (Daw et al. [2006]). Other researchers have placed more emphasis on the role of dopamine in: selective attention (Nieoullon [2002]), the modulation of decision making (Doya [2008]), synaptic plasticity and working memory (Durstewitz et al. [1999], Jay [2003] and Kennerley and Wallis [2009]), the interaction between contextual novelty and reward representations (Guitart-Masip et al. [2010]), impulsive behaviour (Pine et al. [2010]), the temporal precision of reward prediction (Fiorillo et al. [2008]), high-level cognition (Cools [2011]), learning and motivation (Shohamy [2011]) and decisions under reward uncertainty (Huettel et al. [2005], Tan and Bullock [2008] and Schultz et al. [2008]). Through ideas that dopamine signalling underpins beneficial behavioural

decisions made by an agent, we have seen a crossover to economics (Caplin and Dean [2008]), since such signals may be deemed to have some correspondence to to the economic concept of Marginal Utility (Pine et al. [2009]). This field has assumed the label Neuroeconomics (Schultz [2008]).

In addition to these conceptual shifts in the role of dopamine, we have also seen an increase in the breadth of research, such that dopamine is now implicated in feeding behaviours (Narayanan et al. [2010] and Palmiter [2007])and in a developmental role in the formation of the retina (Reis et al. [2007]). Other studies seek to integrate our knowledge of the role of dopamine in the basal ganglia, with other brain systems such as the hippocampus (Kumaran and Duzel [2008]).

Our goal is to arrive at simple circuits that capture essential features of the role of dopamine. Whilst it is useful to have a perspective on the varied interpretations of this role, at least for the purposes of the research described in this thesis, we are more interested in seeking to capture some of the key attributes than we are in explaining its role in all its complexities. For this reason, we take the approach of going back to basics, concentrating on the initial findings of Schultz and his co-researchers (Schultz et al. [1998]). These key attributes are described in more detail in Chapter 7. This has the advantage of avoiding the need to pre-judge which interpretation is the most accurate at this stage. We can simply seek to capture some of the essential characteristics of dopamine signalling, and generate circuits based on these models. Once we have done this we are in a position to revisit these alternative explanations which we do in Chapter 10.

6.4 Integrative Neurocomputational Models

From our perspective, we are limited in the degree of complexity of the computational model of reward anticipatory circuitry that we are able to use. It has to be simple enough so that ultimately we are able to encode features of the circuit into a genome that we can evolve using evolutionary algorithms. This precludes us adopting highly detailed models that, whilst they may be biologically plausible, are less tractable to parametric encoding. Nevertheless, these computational models are able to provide us with some perspectives on features of the system that we should be seeking to model, and provide us with a basis for comparison.

Despite extensive research attempts to model the basal ganglia, we have yet to arrive at a coherent orthodoxy as to the functional significance of its constituent parts (for a review of models see Cohen and Frank [2009] and Chakravarthy et al. [2010]).

One of the more developed of the neurocomputational models is the PVLV model (Hazy et al. [2010]). This model comprises two key elements. The first is a Primary Value (PV) system which governs how dopamine neurons respond to a reward, and a Learned Value (LV) system, which governs how dopamine cells respond to a conditioned stimulus. We shall come back to this model later in the thesis when we compare the results of our model with some of the insights gained from theirs. The development of the PVLV model is an extension of earlier work conducted by O'Reilly (O'Reilly and Frank [2006]). This model focusses on the role of the striatum (in the anticipation of rewards), thalamus (in the failure of reward anticipation), the amygdala (in learning the value of stimuli that may correspond to reward predictive stimuli) and the hypothalamus (as the origin of the Primary Reward signal). The scope of this model is comparatively broad compared to other models that focus more closely on the striato-thalamo-cortical loop (see Humphries and Prescott [2010] for a similarly broad review)

Another sophisticated model is presented by Joseph (Joseph et al. [2010]), which he refers to as the ACE (Actor-Critic-Explorer) explorer model. This is of interest since it emphasizes the potential role of the subthalamic nucleus (STN) and the globus pallidum (GP) in the "exploration" mode, typical and characteristic of many Machine Learning solutions to the value assignment problem. This is not something that is specifically addressed by the PVLV model as it offers a complementary perspective.

Novel models that deal with the integration of the basal ganglia with other brain systems include the role of the basal ganglia in memory retrieval associated with rewarded visual memory tasks (Vitay and Hamker [2010]), action selection and refinement in subcortical loops (Houk et al. [2007]), and a model that focusses primarily on the striatum (Grosse-Wentrup and Contreras-Vidal [2007]). Reinforcement learning oriented models (for a review see Niv [2009] and Samson et al. [2010]) include; the role of the cortex and striatum (Frank and Claus [2006]), hierarchical reinforcement learning (Ito and Doya [2011]), and the development

of pre-frontal cortex representations (Reynolds and O'Reilly [2009]). In Chapter 6 we briefly discuss reinforcement learning and how it relates to models that are proposed in this thesis, but we also point out that Karl Friston's Free Energy Model (Friston and Stephan [2007]) also has some relevance.

Chapter 7

A Hypothetical Reward Adaptation Machine

7.1 Introduction

Figure 7.1 helps to put this chapter in context. Our over-riding goal is to determine or devise models of neural networks with complementary characteristics. These networks are intended to comprise parts of a functional 'vocabulary' necessary to support complex adaptive behaviour. New networks have to satisfy two criteria; they should be tractable to evolutionary search and biologically plausible. Devising such networks is conceptually difficult. Our approach is to break this problem down into simpler sub-tasks. Throughout this thesis we are intentionally taking alternate and complementary perspectives, mechanistic, biological, evolutionary, holistic and deconstructionist, so that we might understand how a primitive circuit might evolve into one which has some of the functional characteristics of a developed brain.

Throughout this thesis we intermittently make use of *mechanistic* models; either as working models (such as in chapter 5) or hypothetical models, such as the case in this chapter. In this chapter for example, we make absolutely no attempt to conceive of this model as one which is tractable to extension or complexification through an evolutionary algorithm. Rather we try and imagine how existing neural networks could conceivably work as a reinforcement learning machine,

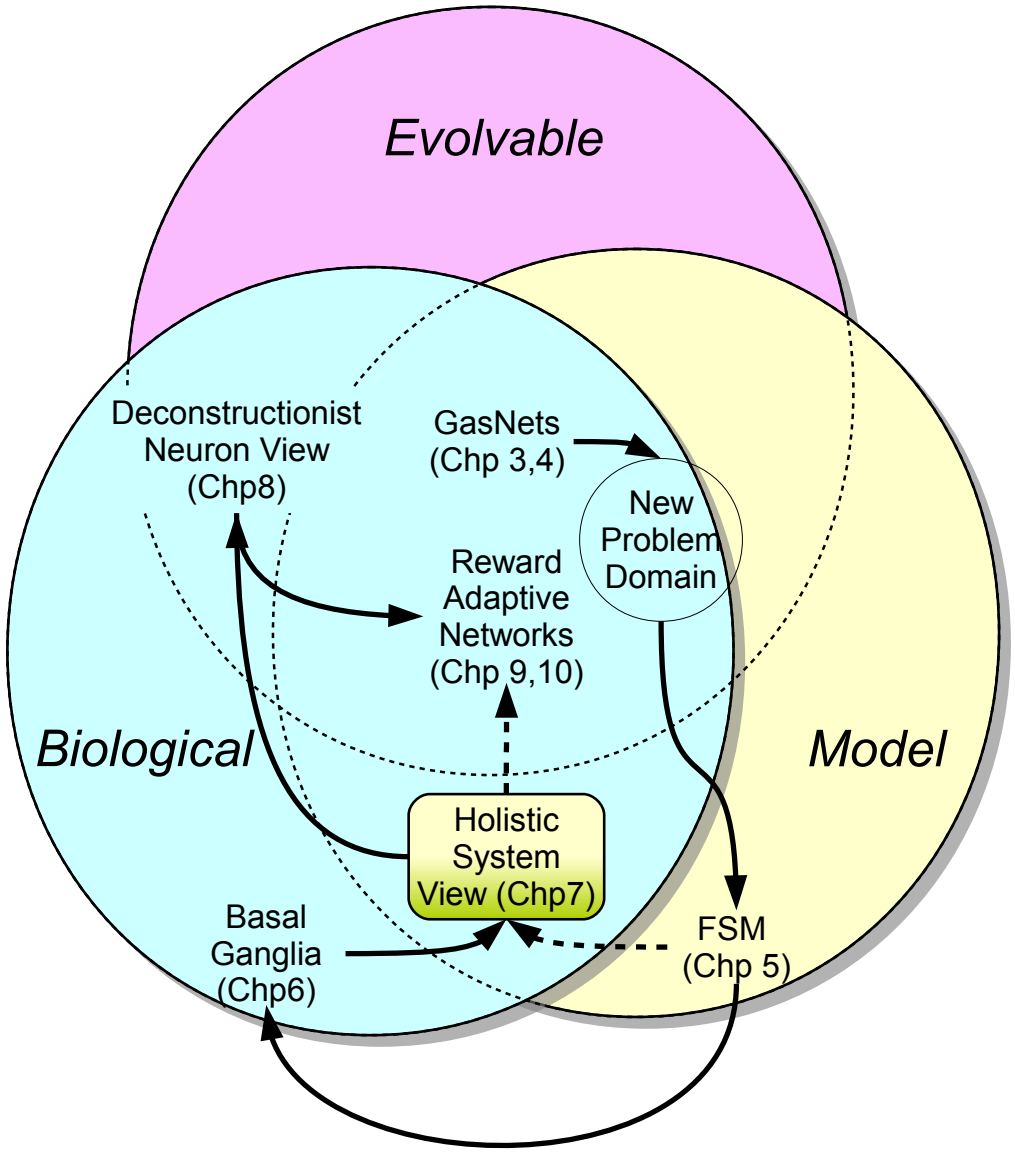


Figure 7.1: **A Concept Map for the Role of this Chapter:** We are attempting to devise new models that are biologically plausible and tractable to evolution. This chapter describes a non-evolvable model that is intended to correspond to a hypothetical *target* for evolutionary search. Through a holistic design we try to identify the necessary attributes that we might wish to see emerge from the evolution of more primitive circuits.

such that the result bears some relationship (albeit remote) to the functional elements of known biological systems. It is intended that this similarity is attained through seeking emulation of the biological systems at two levels. Firstly, at the macro-level it should behave in a fashion that is comparable to classical conditioning. Secondly, at the micro level changes in the system should be elicited by signals that are analogous of those that are known to exist in biological systems (specifically those of the dopamine neuron responding to rewards).

7.2 Identifying Key Attributes

The next two sections describe in more detail the elements that we are seeking to emulate at both the macro-level and micro-level.

7.2.1 Macro-Level: Psychological Conditioning

Our hypothetical reward adaptation machine is to intended as mechanistic analogue of the biological systems that underpin reward adaptation. The machine (if it were to be constructed) once constructed should behave, as far as an external observer is concerned, in a fashion that resembles biological entities in one key area; that of psychological conditioning.

The field of psychology differentiates between two kinds of learning that are termed *classical* and *instrumental* (or *operant*) *conditioning*. Classical conditioning refers to the modification of behaviour that occurs due to the recognition of a stimulus (*conditioned stimulus*) which consistently precedes an *unconditioned stimulus* associated with an *unconditioned response*. After a period of *conditioning* an animal or human gradually responds in a fashion to the *conditioned stimulus* as if it is integral with the *unconditioned stimulus*. In Palovs experiments, that led to the emergence of this field of research, a dog became conditioned to salivate (the *unconditioned response*) at the sound of a bell (the *conditioned stimulus* CS) which occurred prior to the presentation of the food (the *unconditioned stimulus* - US). The second kind of learning is that of *operant* or *instrumental conditioning*, where behaviour is modified due to the presentation of a reward or a punishment following a response.

The model presented here does not differentiate between classical and operant conditioning in its implementation. Both kinds of learning are demonstrated through the modification of behaviour due to the presentation of a reward. Note that the model does seek to incorporate all aspects of classical and operant learning, since it does not address behaviour modification that occurs in response to the presentation of a negative stimulus presented as the unconditioned stimulus, or a punishment in response to a stimulus that we wish to diminish. Although the model does not differentiate between classical and operant conditioning, it does capture aspects of both, in response to the presentation of a positive stimulus (as the unconditioned Stimulus) or the presentation of a reward following a response that we wish to encourage.

The part of the model that emulates classical conditioning is inspired by the activity of dopaminergic neurons (in the basal ganglia) in response to the presentation of a trigger stimulus (CS) prior to the presentation of a reward (US). This is based on observations made by Wolfram Schultz (Schultz [1997]). Various computational models have been proposed to model classical conditioning, since that proposed by Rescorla and Wagner (Rescorla and Wagner [1972]), including those by Grossberg and Schmajuk (Grossberg and Schmajuk [1989],Schmajuk and DiCarlo [1991]) and Buhusi (Buhusi [1999]). The proposed model bears some similarities to these earlier works, but is intended to be simpler, and is not intended to model all aspects of classical conditioning that are observed in the experimental results with humans and animals.

7.2.2 Micro-Level: Brain Signalling

If our hypothetical reward adaptation machine is to provide us with an insight into biological systems, then it should rely only on analogues of signals in biological systems to elicit change. No change should take place in the system, unless it is elicited through a signal that has some biological counterpart. The signals in which we have a particular interest include; dopamine signalling in the basal ganglia (e.g. Schultz [1997]); novelty response in the perirhinal cortex (Bogacz et al. [2001]); and tonic anticipatory activity in the cortex and striatum Suri and Schultz [2001]. Florentin Wörgötter and Bernd Porr have written an excel-

lent review of how these signals are implicated in temporal sequence learning, prediction, and control (Wörgötter and Porr [2005]). We shall seek to generate comparable signals in our hypothetical mechanism, and in so doing imply the existence of certain functionality entities in the brain. On our model these signals are regarded as certain measures of key metrics of system state. Both singly and in combination these signals elicit changes in model state, that correspond to the workings of the mechanism.

7.3 Conceptual Model

Our reward adaptation machines comprises three features (described in more detail in the following sections);

- Reward discovery and expectation mappings (see figure 7.2).
- The construction of action chains (see figure 7.3).
- The readjustment of expectation mappings due to state prediction errors (see figure 7.4).

7.3.1 Reward discovery and expectation mappings

A common formulation for reinforcement learning problems start with the *Bellman's equation*. One of the ways that it expressed is shown in equation 7.1 (Bellman [1957]);

$$J^*(i) = \min_u E[g(i,u,j) + J^*(j)|i,u] \text{ for all } i. \tag{7.1}$$

Typically we imagine a graph where an agent exists at a node corresponding to the current state i, and the agents selects a control u that determines the path selected to arrive at the next state j. The cost of selecting this path is given by the function $g(i,u,j)$ and $E[*|i,u]$ corresponds to an expectation operator. The choice of control u is selected so as to minimize the cost of the next path selected and the costs associated with subsequent optimal selections in a recursive fashion.

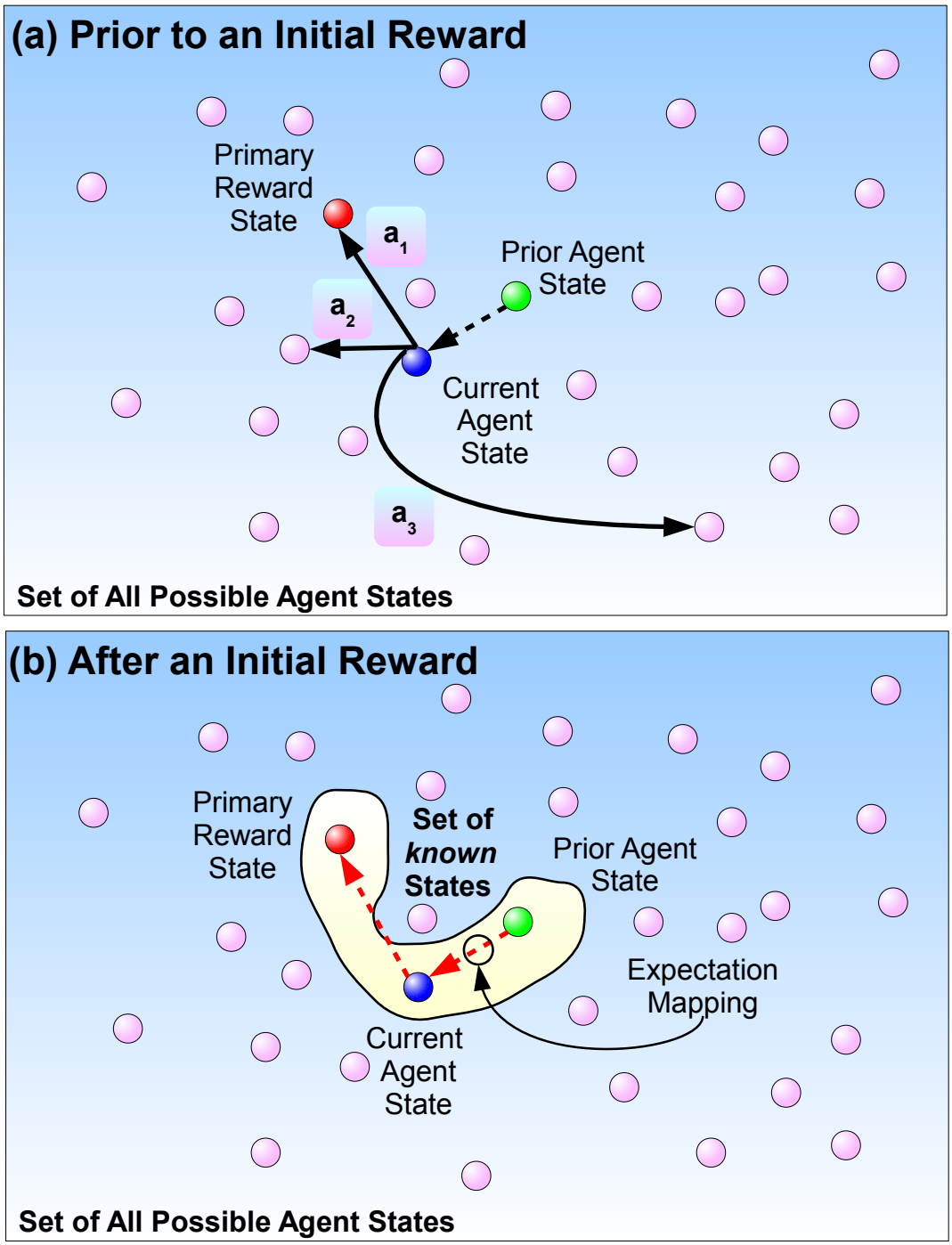


Figure 7.2: **Agent States and Action Graph: (a)** Prior to discovering a reward the agent randomly explores alternative paths (e.g. a_1, a_2, a_3). **(b)** In discovering a reward state, the agent internalizes prior path selections through two means. The first is through storage of the path choices, which results in them being *known*. The second is through some value that corresponds to an expectation mapping, for each path selection that led to the reward state.

This particular formulation of Bellman's equation is taken from Bertsekas and Tsitsiklis' book entitled 'Neuro-Dynamic Programming' (Bertsekas and Tsitsiklis [1996]). Although we do not use this approach directly, the idea of a graph, where each node corresponds to a position in state space and each path corresponds to a feasible actions that might be taken by an agent at a node is quite a useful formulation.

In our model, rather than considering costs that are incurred at each node transition (associated with a path selection), we are more interested in how a mechanism that determines control selection u might adapt its behaviour in a fashion that supports the exploration of state space and the discovery of primary rewards. We use the term primary rewards to refer to exogenous rewards attained by the agent through the attainment of some terminal state.

Prior to discovering a reward the agent randomly explores alternative paths (e.g. a_1, a_2, a_3) as illustrated in figure 7.2(a). In discovering a reward state, the agent internalizes prior path selections through two means. The first is through storage of the path choices, which results in them being *known*. The second is through some value that corresponds to an expectation mapping, for each path selection that led to the reward state (see 7.2(b)). When we refer to an expectation mapping, we imagine a value that is associated with the action selection choice taken at a given point in state space and the subsequent outcome of this choice (i.e. the next point in state space). It is the value of this expectation mapping is used in determining future action choices, when the same point in state space is rediscovered. The purpose of internalizing this state data is to bias future path selections such that the likelihood or rediscovering the reward is increased.

7.3.2 The construction of action chains

The discovery of a single pathway (for example, a sequence of motor selection choices to a reward from a point in state space) that leads to a reward whilst beneficial, may be sub-optimal. We wish to ensure that alternative pathways are also explored, and alternative reward paths also internalized. This is supported by an enlargement of the set of *known* states, and the storage of a values correspond

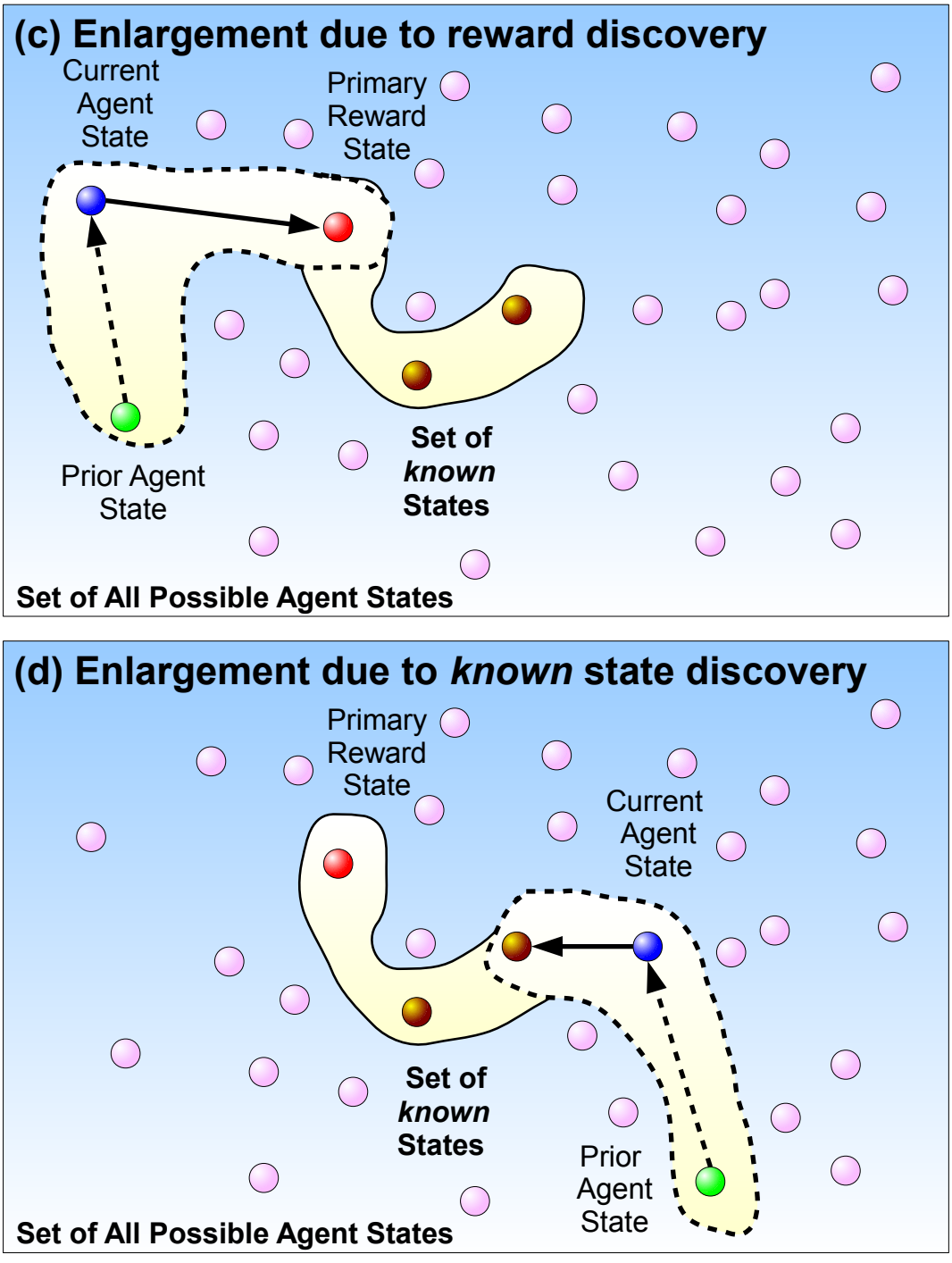


Figure 7.3: ***Known* State Enlargement:** There are two means by which new reward paths are internalized; **(c)** through the discovery of an alternate reward path, or **(d)** through the discovery of a *known* state from an *unknown* state.

to action-selection expectations corresponding to this new path (as illustrated in figure 7.3(c)).

A node in state that is *known*, is that which has through past exploration been discovered to be a node on a path that is expected to result in a reward. Nodes in state space that are not *known*, are by definition ones which have yet to be demonstrated as comprising a node on a path leading to a reward. However, we should like to incorporate some mechanism by which states that are currently *known*, when discovered from an arbitrary node in state space that is not *known* are incorporated into the path. This allows us to extend the length of path, and include states that may bear some causal relationship to the final reward outcome (as illustrated in figure 7.3(d)). Note that this latter facility corresponds to the phenomena that we observe in conditioning where dopamine activation signals migrate to the earliest occurrence of features that show a correlation with the final reward.

7.3.3 Readjustment of expectation mappings

Finally our model should be robust to errors. It should undergo some re-adaptation if a path that had previously been determined to lead to a reward no longer appears to do so. Without this re-adaptation the agent would be biased to repeating action selection choices after only one successful occurrence. There are two mechanisms by which we imagine that such re-adaptation could take place (illustrated in figure 7.4). The first is through an adjustment in expectation mappings. Note that it is the expectation mapping value that biases subsequent action selection choices. By reducing this value we are reducing the probability that a given path will be taken. Over a prolonged period of time, we should also support a second mechanism whereby after repeated failures, the particular path that was previously shown to result in a reward is removed from the set of *known* states. This will ensure that the subsequent discovery of these states from *unknown* states, does not lead to an enlargement of the path.

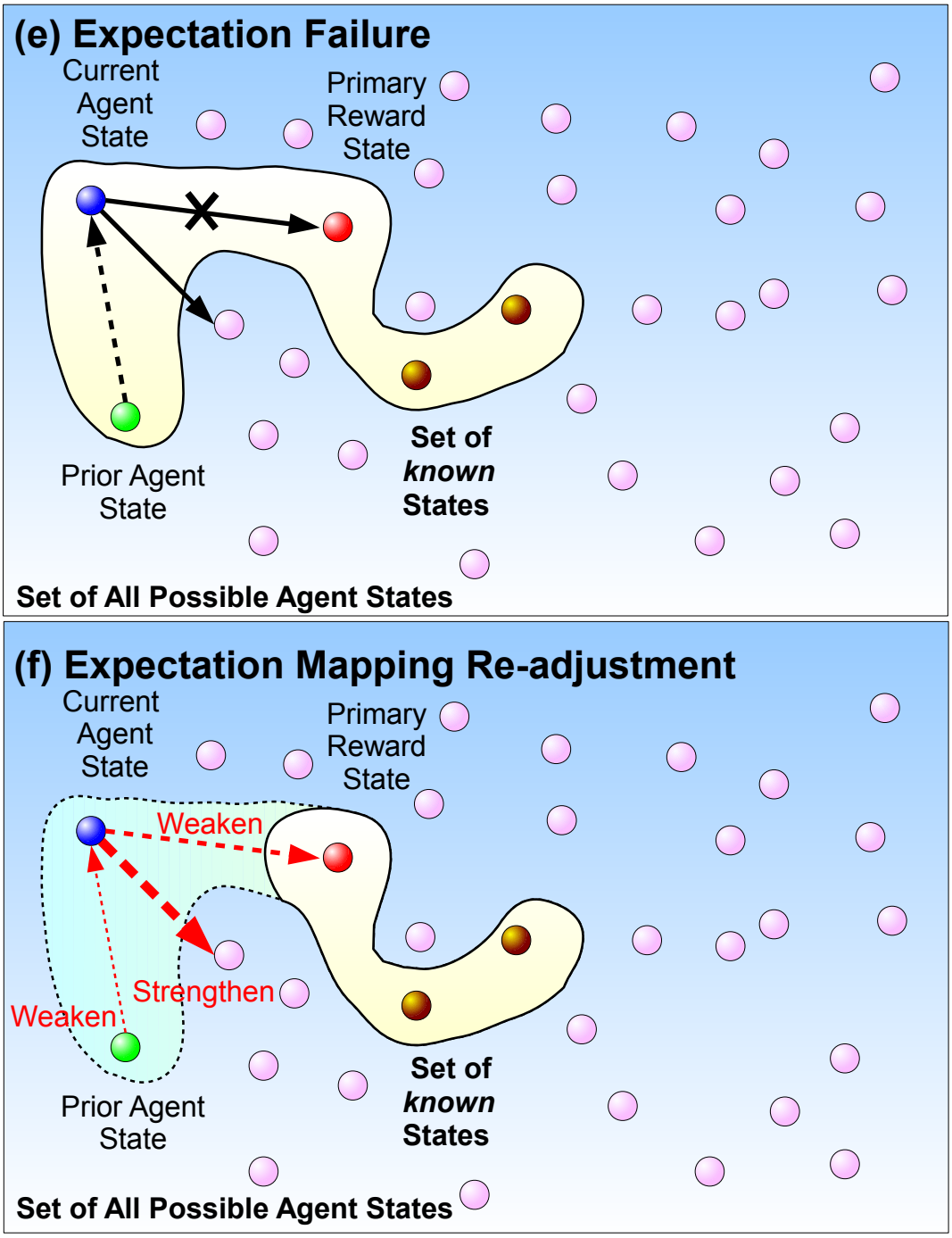


Figure 7.4: **Readaptation on failure:** **(e)** In circumstances where the expected outcome of an action selection taken at a given state does not result in the expected outcome, we need to **(f)** re-adjust expectation mappings. This involves reducing the value of the expectation mapping for the failed outcome, and increasing it for the new outcome. Ultimately a path that has been discredited repeatedly should be removed from the set of *known* states.

7.4 Machine Architecture

The features that we would like to see in our hypothetical reward adaptation machine have already been elucidated. Our challenge is to conceive of how such a system might be implemented through the configuration of conventional neural networks. One possible solution is illustrated in figure 7.5. It illustrates the key components of the mechanism, and indicates the signals that are used to elicit change in the system. In this section we outline the key elements of its architecture. In section 7.6 we shall describe how it works from a functional perspective. There are four key components; a Sensor memory bank, a SOFM of sequence patterns, an expectation mapping array, and a motor program activation switch. There are three signals in the system that correspond to measures of the extent to which a sensor state is; *novel*, *known*, and *expected*. We also assume the existence of signal indicative of a *delay* in an expected state.

7.4.1 Sensor Memory Buffer

The first element in our design is that of a short-term memory buffer. The role of this memory is to record the state data (propriceptive and exteroceptive) associated with recent actions and their outcomes. The construction of the memory buffer represents rows of state data, where each row corresponds to a different node in the state graph. When we transition from one node to another, the sensor data is passed along the buffer in a fashion similar to a pipeline. When an action is taken all data is shifted along the pipeline prior to the new state data being written to the top row. This data is transient. The contents of this sensor memory buffer are processed either when a primary reward has been detected, or a *known* state is discovered from an *unknown* state.

7.4.2 SOFM of temporal sequences

We us a self-organizing feature map (SOFM) is to store pattern sequences. When a primary reward is detected (or a *known* state discovered from an *unknown* state), the entire contents of the sensor memory buffer are deemed to correspond to a sample vector that is presented to the SOFM. In this way the SOFM is used

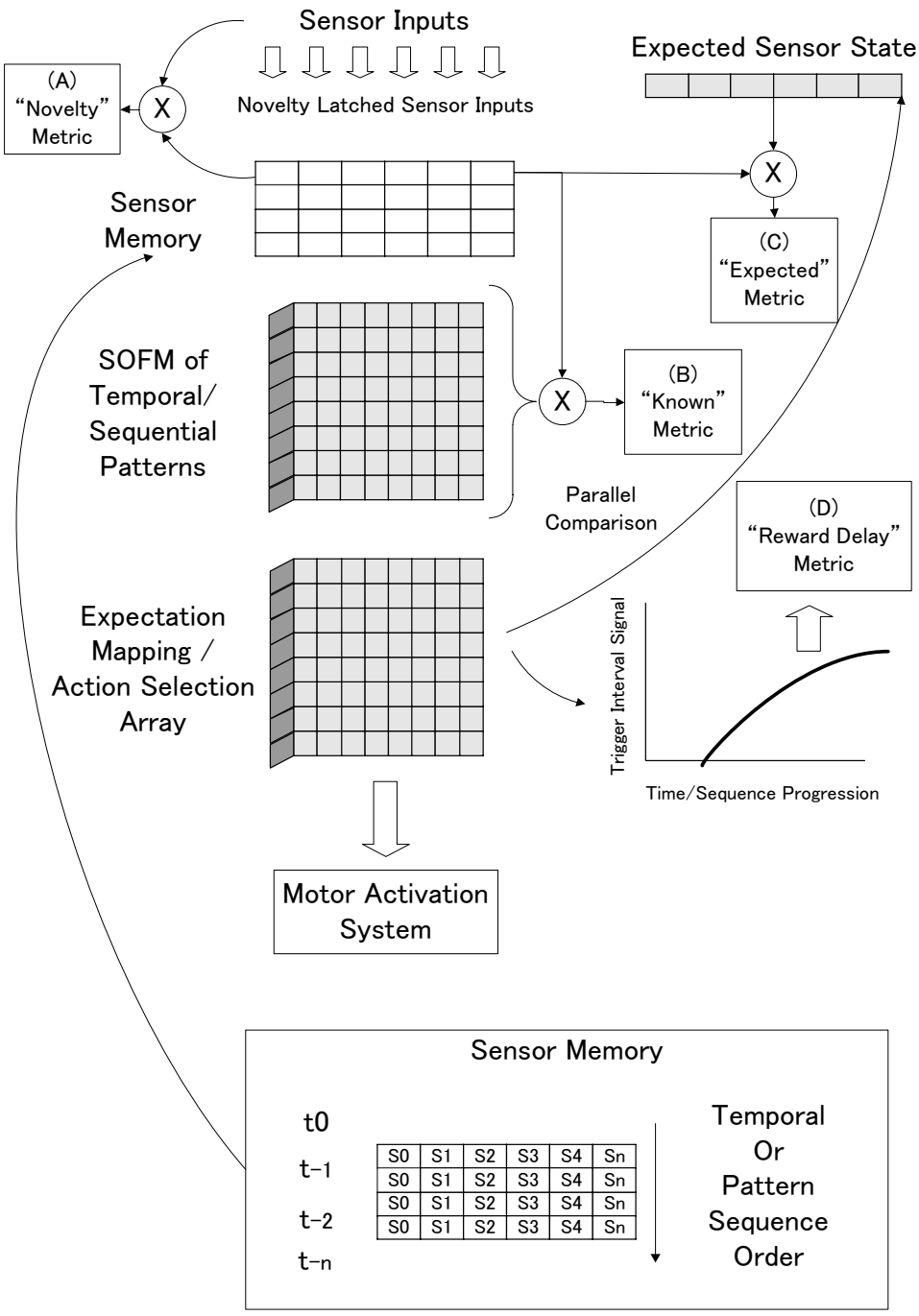


Figure 7.5: **Machine Architecture - Overview:** There are four key components. (1) *sensor memory*, organized to store sensor data in a temporal or pattern sequence order. (2) A SOFM which stores pattern sequences (the winning pattern corresponds to the current *known* state). (3) A corresponding expectation mapping array (where each cell of the SOFM has a corresponding entry). (4) Motor program activation switch

only to store pattern sequences that are directly associated with rewards, or the discovery of states that correspond to pattern sequences that have in the past lead to rewards.

7.4.3 Expectation mapping array

For each entry in the SOFM, there exists a corresponding entry in the expectation mapping array. The function of this array is to determine, given the current winning vector in the SOFM, the next node in the state graph. Note that the values associated with an expectation mapping, determine the probability that certain actions will be repeated (since motor actions are stored as proprioceptive values on the sensor memory buffer). Since only path sequences are stored when they have previously resulted in the attainment of a reward, this biases the agent into repeating those actions.

7.4.4 Motor program activation switch

The SOFM of temporal pattern sequences has a 'winning' entry that corresponds to the nearest *known* state to the current sensor state. This in turn allows us to identify the expectation mapping identifying the next states, and the motor signals necessary to activate the appropriate control values. It is the role of the motor program activation switch to ensure that these motor control signals are used to activate and deactivated low-level motor control programs.

7.5 Machine Internal Metrics

Figure 7.5 shows an over-all schematic of the model. There are four important metrics shown that are used internally in the model to elicit changes in the system when certain criteria are reached. These metrics are described in the following sections. In the model that we present here, we are not particularly interested in specifying the exact functions that implement these metrics. Our hypothetical model is intended as a class of systems that may have alternative functional forms, but yet operate in the same general fashion (as outlined in this chapter).

We shall however give examples of functions that might implement such metrics for the purpose of explaining how we might expect such a machine to operate.

7.5.1 Novelty Metric

We describe a buffer for sensor memory in section 7.4.1. In the schematic shown in figure 7.5 we illustrate the buffer as containing samples of environmental and proprioceptive data at discrete intervals. Our model for this buffer is as a pipeline of sensor data, where new data enters at to top and old data is pushed out the bottom. The first slot in the buffer corresponds to sensor data that was latched into the buffer, on the last occasion that a significant state change was detected. Since we imagine a mechanism of finite memory, comprising discrete sensor samples, we have to make a design decision regarding when or under what circumstances new sensor states should be latched into the buffer (i.e. all buffer entries pushed down the pipe, and the current). We do so based on the a measure of the extent to which current sensor values are *new* or *novel* compared to the most recent data in the first slot of the buffer. We imagine the existence of a novelty metric (or novelty function) of the form;

$$y_N = \|v_0^S(t) - v_1^S(t - \tau_1)\| \tag{7.2}$$

where y_N corresponds to a novelty value, $v_0^S(t)$ corresponds to an array of environmental (and proprioceptive) sensor values at the current time, and $v_1^S(t - \tau_1)$ corresponds to the prior sensor values latched into the 'most recent' slot of the sensor buffer. When this *novelty* value corresponds to a measure of the euclidean distance between the current sensor state, and that which is regarded as our most recent memory. When this exceeds some arbitrary value, current sensor data is latched into the buffer.

The decision to use of novelty rather than simply the passage of time is based on two ideas; firstly, significant events may take place sparsely over time, secondly, it allows us to segment continuously varying data into equally spaced chunks. An ideal system would only latch in those sensor states that corresponded to nodes in the space of significant action states (illustrated in 7.2). However, we do not know in advance what states are significant. We can conceive of situations where no

observed state change occurs, but some action is required after some interval. To support such situations, we would envisage an internal indicator that integrates the passage of time, such that when a certain 'distance' has been reached (i.e. time has elapsed), then this is treated *as* a state change.

A new sensor state is latched into memory as long as it exceeds a certain novelty threshold. This threshold itself could be variable, corresponding to some function of arousal or sensitivity (as it is signalled in the brain by serotonin for example, e.g. Jing et al. [2008]). Another reason for incorporating the notion of *novelty* into our model, is that there is wide-spread evidence to suggest that novelty plays a significant role in learning in biological systems (Xiang and Brown [1998], Bogacz et al. [2001], Wittmann et al. [2008], Guitart-Masip et al. [2010]).

7.5.2 Known Metric

In addition to this short-term memory buffer, the model has a self-organizing feature map (SOFM), which stores sequences of state data that are associated with a chain of events leading to a reward. For this reason we refer to this SOFM as the SOFM of *reward state-sequence patterns*. The *known* metric corresponds to some measure of the extent to which a given entry in the short-term memory buffer resembles an existing entry in the SOFM. Note that it is only when a chain of events leads to a reward, that we update the SOFM. So the very fact that an entry in the sensor buffer is *known*, implies that it is part of a reward sequence. The metric for determine whether or not a current state is *known* may take the form of some minimum distance between the *current* sensor state, and all entries in the SOFM;

$$y_K = \min_i \|v_0^S(t) - v_i^K\| \text{ for all } i \tag{7.3}$$

where y_K is the corresponds to the *known* value, $v_0^S(t)$ corresponds to data in the most recent sensor slot in the sensor buffer, and v_i^K corresponds to the sensor data stored in the SOFM at cell index i.

We originally described the SOFM of *reward state-sequence patterns* as being initialized with random small valued vectors. However, an alternative implementation might store pre-determined patterns that are associated with terminal

rewards. Such entries could be genetically determined. Such a sensor pattern might correspond to the eating food for example. Random exploration of the environment would ultimately lead to the discovery of this predetermined sensor state. The very first time a sensor state latched into the short-term memory buffer that corresponds to this state would be detected as *known*. Since a predetermined entry in the SOFM already exists. However, since this is the first time that this state is detected as *known*, the corresponding entry in the *expectation mapping array* would indicate, that this state is *unexpected*. The discovery of a *known* state, from an *unexpected* generates a trigger signal. The trigger signal elicits functionality whereby the contents of the short-term memory buffer are latched into the SOFM of *reward state-sequence patterns*, and the *expectation mapping array* is updated accordingly. Since this functionality is a significant aspect of the model, we deal with this in more detail in the following section.

7.5.3 Expected Metric

Each entry in the SOFM has a corresponding cell in the *expectation mapping array*. This cell has three elements. The first in an indicator of the index of the SOFM entry that is expected to *proceed* the current state. The second is a measure of the *strength* of this expectation. The third is an indicator of the index of the SOFM entry that *precedes* the current cell.

If an entry in the sensor buffer is *known*, then by definition there exists a corresponding entry in the SOFM of *reward state-sequence patterns*. This entry will in turn, have a corresponding cell in the *expectation mapping array*. Since this cell stores the index of the SOFM entry that precedes the current *known* state (from which we are also able to determine an *expectation strength*), we are able to determine whether or not the entry in the sensor buffer that precedes the current *known* state, also matches that of the SOFM cell indicated as that which precedes the current cell (stored in the *expectation mapping array*). In this way, we can determine if any *known* entry in the sensor buffer, is also *expected*.

7.5.4 Reward Delay Metric

The reward delay metric is a used to measure the elapse of novel state sequences after a new input to the sensor buffer has been determined to correspond to a *known* state. If the current *reward delay value* exceeds a certain threshold value, *without* a sensor data transition to the new expected state (as determined by the *expectation mapping array*), then this is indicative of a *failed expectation*. A failed expectation will then result in an adjustment to the *expectation mapping array* entry for the *known* state.

7.6 Machine Function

Up until this point we have described the structure of the reward adaptation machine and functions for determining key values. It is this section that we give examples of circumstances under which certain value conditions result in change in the machine.

7.6.1 Updating the SOFM

When an unanticipated reward is discovered, we 'unwind' the contents of the sensor buffer and update the SOFM. This process is illustrated in figure 7.6 and figure 7.7.

The *sensor memory* is traversed in a sequential manner, such that each buffer row (or buffer segment) in the *sensor memory* is presented to the SOFM in a retrospective fashion. By this we mean that an arbitrary time/sequential chunk of sensor data in the *sensor memory* is presented as an exemplar to the SOFM. Once this has been completed, and the weight vectors of the SOFM adapted in line with conventional competitive learning, the next oldest time/sequential chunk of sensor data is presented as an exemplar to the SOFM. In this way, the SOFM is updated with sensor patterns that precede a terminal pay-off. All entries in the SOFM are potentially significant as they relate to a chain of events that has resulted in a terminal pay-off.

The determination of the current *known* state (corresponding to the current winner of the SOFM buffer), allows us to determine the *expected sensor state*,

from the corresponding entry in the expectation mapping array. A comparison of the current sensor state and the *expected sensor state* gives us an *expected metric*. If a Primary reward stimulus is detected, coincident with an *unexpected* signal, then circuit adaptation occurs. This comprises two parts. In figure 7.6(1) The current Sensor State is presented to SOFM as a *winning vector*, resulting in SOFM adaptation using the conventional model. Each entry in the SOFM is said to be a *known* state. In figure 7.6(2) We pass through each entry in the *sensor memory* Buffer, determining the corresponding entry in the SOFM buffer. From here we are able to update the Expectation Mapping in the Action Selection Array. The strength of the mapping of consecutive *known* pattern state entries is increased. Since each Expectation Mapping entry is associated with a specific motor pattern, state sequences that result in a Primary Reward are entrained.

7.6.2 Updating Expectation Mappings

A second important adaptation takes place as a consequence of the discovery of a *known* state that is *unexpected*. This is illustrated in figure 7.8. Even if no Primary Reward is present, the discovery of a *known* state from an *unknown* state (i.e. one that does not have a corresponding entry in the SOFM), results in the same adaptation processes that would occur to a Primary Reward. We know that all *known* state entries have been presented to the SOFM, since they constitute part of a reward chain of anticipated state sequences that may lead to a Primary Reward. Discovery of a *known* state from an *unknown* state updates the SOFM, so that the *unknown* state is now *known*. At the same time, expectation mappings between this state and the *known* state are strengthened. The discovery of *known* state from an *unknown* state constitutes an **implied reward**, that results in the same circuit adaptation as a **primary reward**. Since the expectation mappings between the new *known* state and existing *known* states are strengthened, we are effectively lengthening the chain of pattern sequences that are expected to lead to a Primary Reward.

When each sensor data entry in the short-term memory is presented to the SOFM, we are able to identify a winning entry in the SOFM. This entry also corresponds to a cell in the *expectation mapping array*. As we traverse the contents

of the *sensor memory*, and present chunks to the SOFM, we discover the closest entry in the SOFM that corresponds to this chunk. The sequential presentation of these chunks will result in sequential winners that have a defined temporal/sequential relationship with one another. The winning cell in the SOFM (corresponding to an older stimulus chunk) will have an anticipated mapping to the winning cell that follows it (in a temporal or sequential sense). The weights associated with these 'anticipated' sensor state transitions are strengthened through a modified Hebbian process. Repeated presentations of the same winning SOFM cell entry pairs, will thereby result in a stronger *expectation mapping* between the two entries (the results of which are stored in the *expectation mapping array*). This is illustrated in figure 7.7.

Note that in a more complex system the pattern of activation in the SOFM could itself be treated as *sensor memory*. Such a system would maintain hierarchies of SOFM, each of which would hold data corresponding to coarser levels of temporal or sequence granularity. Such a system would have the capacity to store events at different time-scales. Those SOFM buffers which stored data with a longer period, could be used to provide us with temporal *context*. Such a system would be capable of learning reward sequences that relied upon this *context*, in a fashion that would complement recent or short-term state changes.. An attractive attribute of using the SOFM in this fashion, is that we are only presenting exemplars that are known to have a likely temporal/sequential relationship with a terminal reward pay-off. An SOFM coupled directly to the sensor input would be biased towards the statistical representation of all sensor inputs, irrespective of their relationship to a terminal reward. We effectively achieve an efficiency of memory through this model, by only storing sequences that lead to a reward.

A significant feature of the model as presented here, is that there is no differentiation between genetically pre-determined reward states (e.g. those sensor reward patterns that may have been discovered via evolutionary search and constitute pre-coded entries in the SOFM) and dynamically allocated *known* states. Note that when we discover one of these terminal reward states, we transfer the contents of the Sensory Memory into the SOFM, such that states that precede a terminal reward state become *known* states themselves.

The effect of this is that with the discovery of *any known* state, from a state

that has a low *expectation value* associated with it, we then engage in exactly the same process as we would if we had discovered a terminal reward state. That is, we are treating one of the precursor sensor states as if it is itself a reward stimulus state. In the process we latch the contents of the *sensor memory* into the SOFM, update the winners sequentially, and then update the expectation mappings associated with each SOFM entry. In this way we can build up a map of *known* states that precede a reward state, without directly needing to experience the reward.

This potentially has the danger of misclassification. It is conceivable that one of the sensor chunks that was latched into the SOFM has no temporal/sequential significance. We might imagine that the discovery of potential pre-cursor states in the fashion described above should not lead to an adaptation in the SOFM, until it has been determined that the current sensor chain definitely results in a terminal pay-off. Such a modification, whilst making the system more complex, does not make it intractable. However, at this stage it is not intuitively obvious that the model as it described here is insufficient to capture the necessary dynamics that we are seeking.

It may well be the case that the statistical presentation of the non-relevant sensor state to the SOFM as an exemplar would be insufficient, over a period of time, to ensure that it persisted as a significant entry. It should be remembered that the SOFM is updated every time a reward is experienced, and its associated precursor states are also updated.

An important aspect of a robust reinforcement learning system is the capacity to re-adapt to changing environmental circumstances, or to re-learn erroneously discovered expectations. A comparison of an expectation with the current sensor state, together with the inclusion of a value that increases with time as the expected reward fails to appear (the *reward delay* metric), allows us to re-adapt the system to failed expectations. We can do this simply by adjusting the expectation mapping (i.e. weakening the expectation mapping between the current and anticipated states). It is conceivable that the expectation mappings constitute an array of weights to all potential sensor states, such that we choose that which has the highest value when seeking to discover an expected state. The repeated failure of an expected state corresponding to a fixed current state will naturally

result in a decline in the strength of its expectation mapping.

7.6.3 Selection of an action

Up until this point we have neglected discussion of how determination of current state, and its potential to lead to a terminal reward, is tied-in with motor activation.

There are number of alternatives here. Whilst this may appear a rather vague statement, it is actually of benefit to us from an evolutionary robotics perspective. One of the challenges is to imagine how such a system may fit within an evolutionary robotics framework.

The output of the *expectation mapping array* could be used in a number of ways. In its simplest form it could act as additional input to a pre-evolved or co-evolved neural circuit, making simple inhibitory or excitatory connections. A more powerful implementation could see us using the output of the Expectation Mapping array to select a weight vector which corresponds to the coupling weights of individual sensor-motor pairs. This in itself would be sufficient for us to demonstrate behaviour analogous to that exhibited by a Finite State Machine in the generation of sequential activity, and possibly powerful enough to simulate invertebrate control systems.

An even more sophisticated approach could see us engaging in real-time local search or optimal motor activity. The exciting aspect of this idea is that the reinforcement learning model will potentially break down the reward achievement process into sensor states, with expected sensor transitions discovered through random exploration. Once we have a known current state, and a known expected state, we can engage in a local search of motor space, either to seek a better motor pattern that fulfils expectations (e.g. energy optimization), or to discover other *known* states, preferable to that associated with the current mapping.

7.6.4 Changes due to failed expectations

Note that the process of failed expectation in the model that we describe above will naturally lead to a random exploration of state space in an immature robot. Failed expectations will result in the diminution of weights associated with the

incorrect mappings, resulting in alternative sensor states (an their associated motor actions) being explored.

The last diagram (below) seeks to explain some of the principles incorporated by the described model in a schematic fashion, by considering the state space of all sensor states. Within the state space of all sensor states, we are seeking initially to discover *significant* sensor states, i.e. those that have the potential to lead to a terminal reward. In the process, through repetition, and through the discovery of states that lead to rewards in a consistent fashion, we are able to build up an expectation mapping that in itself allows us to choose an appropriate motor action for a given sensor context.

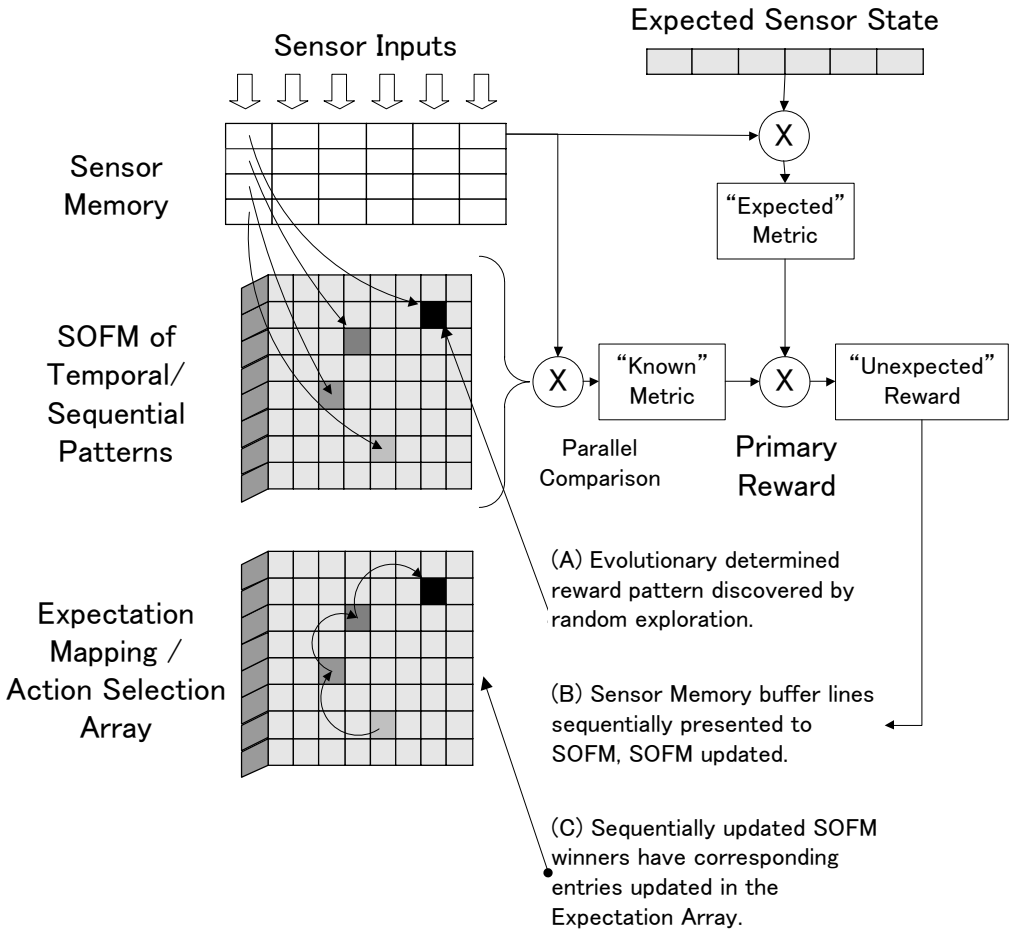


Figure 7.6: **Conceptual Model - Initial Adaptation to An Unanticipated Reward:** The determination of the current *known* state (corresponding to the current winner of the SOFM buffer), allows us to determine the *expected sensor state*, from the corresponding entry in the Expectation Mapping Array. A comparison of the current sensor state and the *expected sensor state* gives us an *expected metric*. If a Primary reward stimulus is detected, coincident with an *unexpected* signal, then circuit adaptation occurs. This comprises two parts. (1) The current Sensor State is presented to SOFM as a *winning vector*, resulting in SOFM adaptation using the conventional model. Each entry in the SOFM is said to be a *known* state. (2) We pass through each entry in the *sensor memory* Buffer, determining the corresponding entry in the SOFM buffer. From here we are able to update the Expectation Mapping in the Action Selection Array. The strength of the mapping of consecutive *known* pattern state entries is increased. Since each Expectation Mapping entry is associated with a specific motor pattern, state sequences that result in a Primary Reward are entrained.

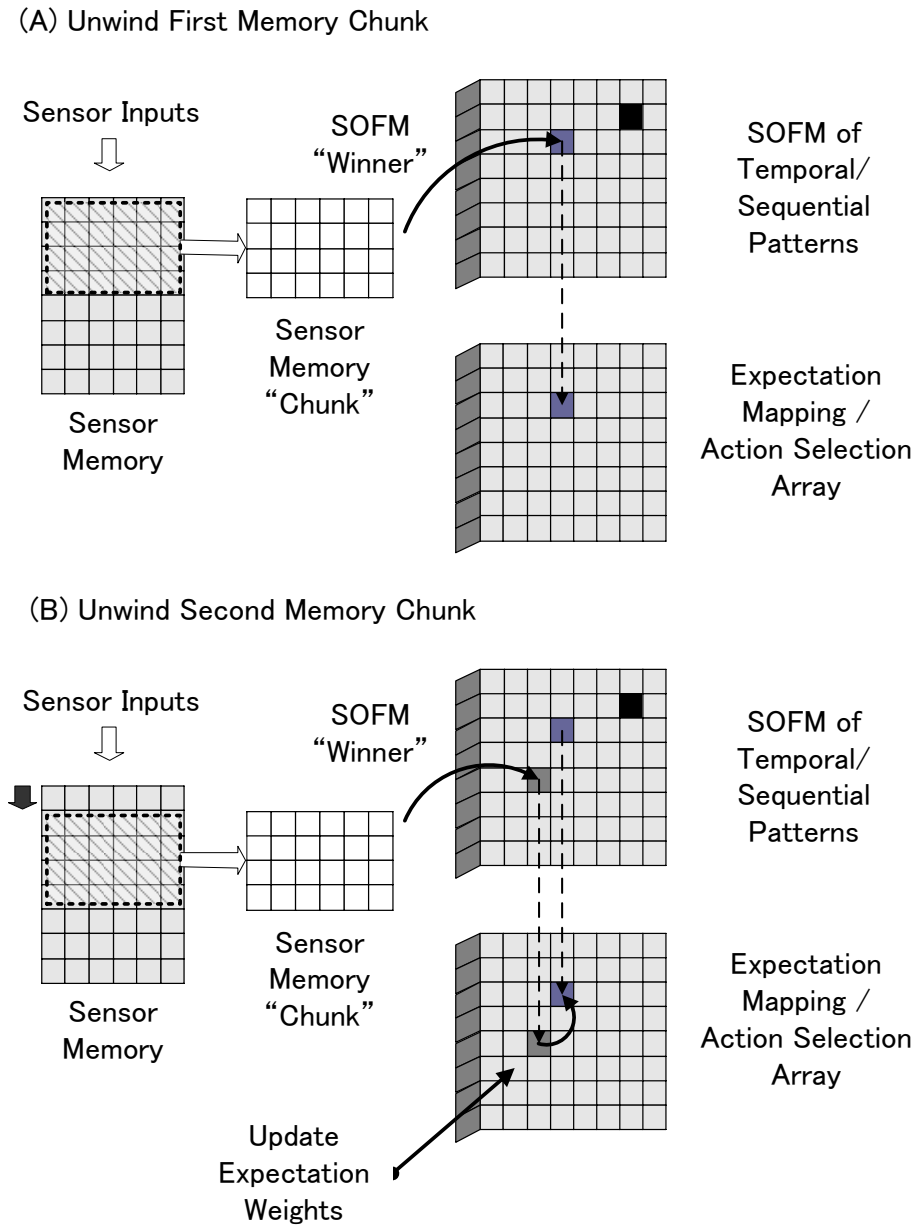


Figure 7.7: **Conceptual Model - Update Expectation Mappings:** This schematic shows in more detail the adaptations that take place when an unexpected reward is detected. Sequentially we *unwind* the contents of the Sensor Buffer, so that we are able to update the strength of the mappings between *known* sensor states that have resulted in this unexpected reward.

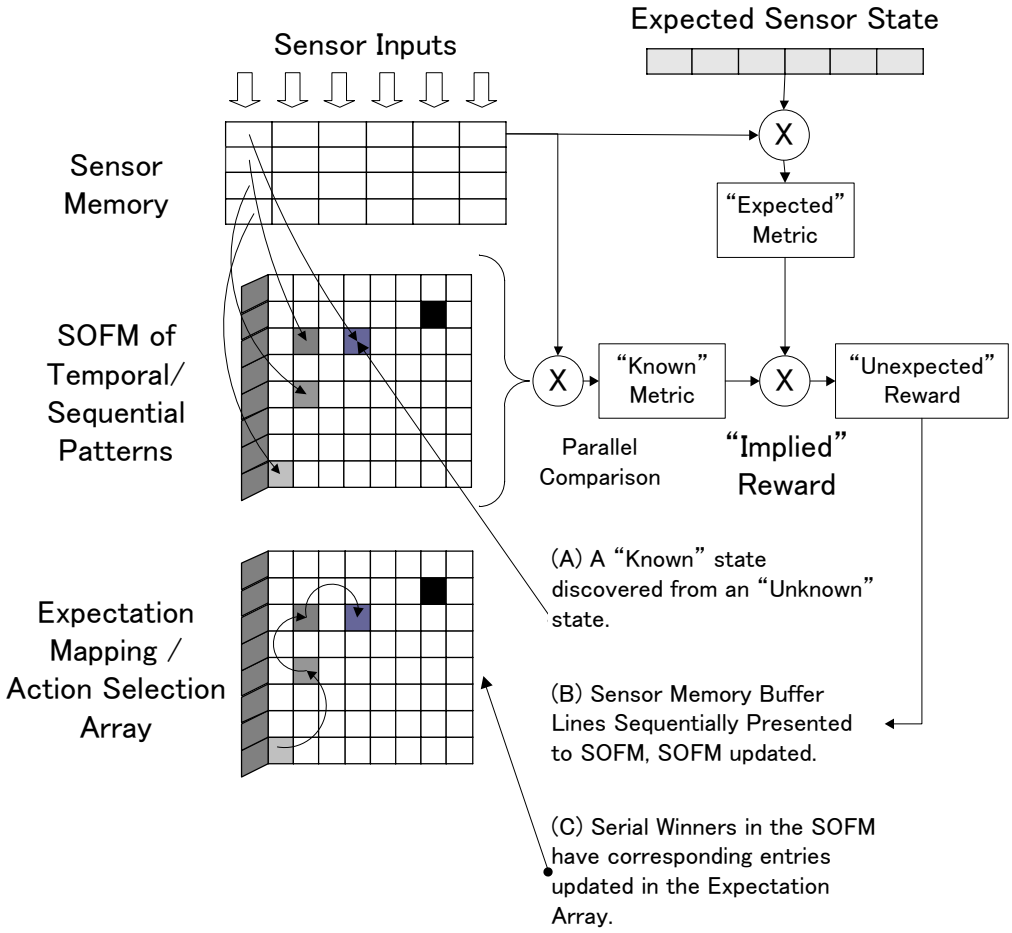


Figure 7.8: **Conceptual Model - Lengthening the Reward Chain:** Even if no Primary Reward is present, the discovery of a *known* state from an *unknown* state (i.e. one that does not have a corresponding entry in the SOFM), results in the same adaptation processes that would occur to a Primary Reward. We know that all *known* state entries have been presented to the SOFM, since they constitute part of a reward chain of anticipated state sequences that may lead to a Primary Reward. Discovery of a *known* state from an *unknown* state updates the SOFM, so that the *unknown* state is now *known*. At the same time, expectation mappings between this state and the *known* state are strengthened. The discovery of *known* state from an *unknown* state constitutes an **Implied Reward**, that results in the same circuit adaptation as a **Primary Reward**. Since the expectation mappings between the new *known* state and existing *known* states are strengthened, we are effectively lengthening the chain of pattern sequences that are expected to lead to a Primary Reward.

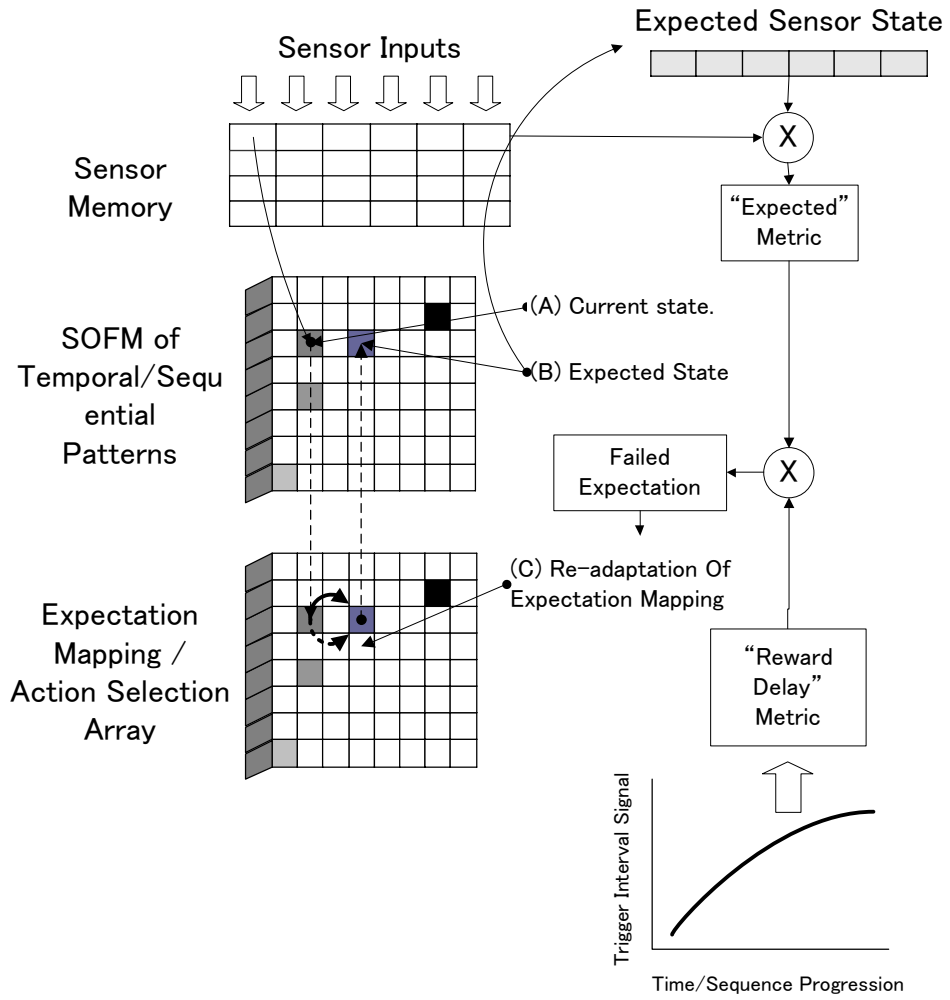


Figure 7.9: **Conceptual Model - Adaptation To A Failed Expectation:** Under any circumstances where motor actions taken in a *known* sensor state, do not result in the expected *known* sensor state, expectation mappings are weakened. In a sophisticated system, the trigger for this adaptation would take place after a certain subjective time period had elapsed. Note that we use the term **subjective time** rather than absolute time, since this may relate to the passage of event patterns that are deemed to be relevant. For example, if a reward was gained through pushing a button to open a door, the absolute time period taken for the door to open may not be significant. If after pressing the button, a window was to open, this might be interpreted as a significant event that corresponds to a increment in the passage of qualitative time experienced.

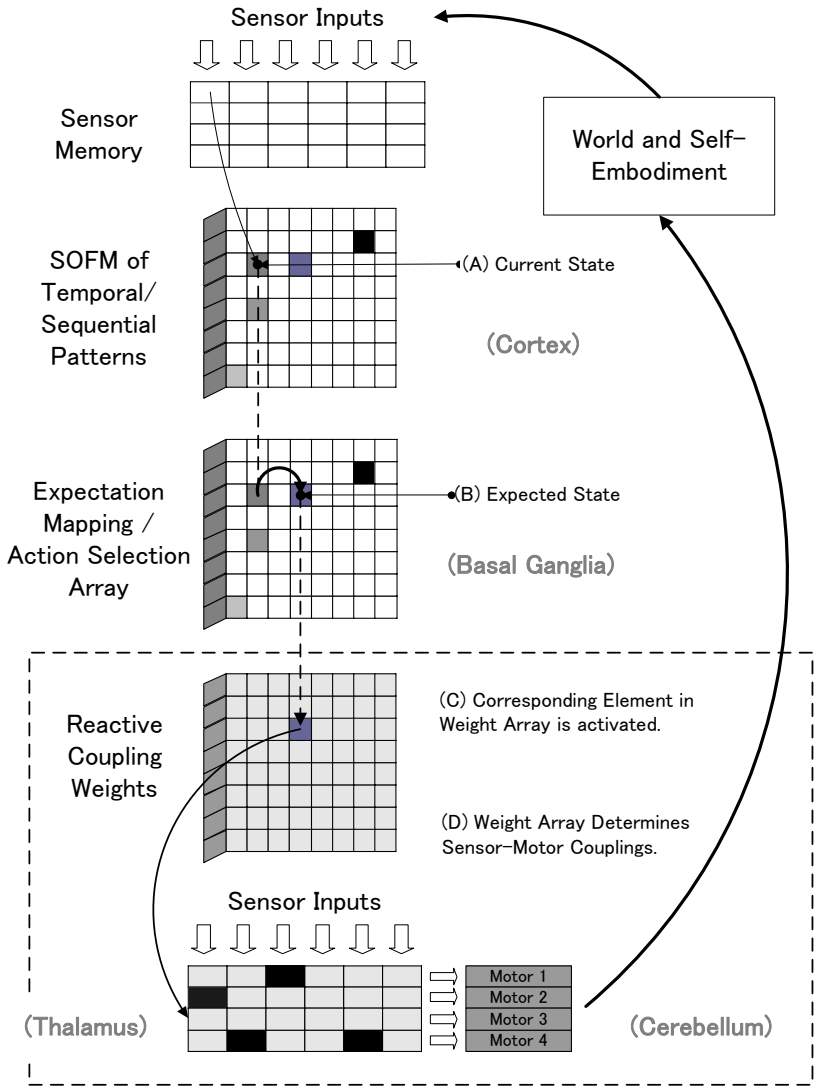


Figure 7.10: **Conceptual Model - Expectation as Action Selection:** Strengthening of expectation mappings occurs due to (1) an unexpected reward and (2) discovery of a *known* state. Weakening of expectation mappings occurs on a failed expectation. If we associate each *known* state entry with a motor command, then we create a closed loop where subsequent activities are based on motor commands that have in the past resulted in the attainment of rewards. The agent 'does' what it 'expects' to do. Since only *known* state patterns that lead to a reward are stored in the SOFM, the detection and execution of motor command sequences that correspond to our expectation mappings can be expected to lead to a reward. We have labelled the components (in brackets) with their nearest functional analogue in the brain.

7.7 Alternative Approaches to Understanding Reward Adaptation

The main motivation of this chapter, was to try and understand the processes of change that take place in the brain, that underpin behavioural complexity as they relate to reward adaptation. Ultimately we wish to devise a model that might have be subject to evolutionary change, in keeping with the evolutionary robotics methodology. An understanding of other perspectives on reward adaptation, that differ from the hypothetical model described here, helps to informs us on these future design decisions. Two areas that are of particular interest are those of *reinforcement learning* and Friston's *free energy model*. Whilst neither of these two topics provide us with a design for the constructing neural networks that might support them, they do provide us with alternative conceptual models for understanding the kind of processes that are deemed to take place in biological systems, and how they complement the hypothetical model that we have outlined here.

Dopamine signalling is closely associated with the reinforcement learning model (Sutton and Barto [1998], see Samson et al. [2010] for a recent review). Specifically dopamine phasic activity is said to correspond to the prediction error in the Temporal Difference approach to reinforcement learning.

In recent years however, we have seen the emergence of a 'grand theory' of brain organization in the form of the *free energy model* (Friston and Stephan [2007]). Somewhat confusingly, 'free energy' is used in an information theoretic sense, in a fashion that corresponds to prediction error. The emphasis on prediction and anticipation, and a system wide perspective, potentially offer a formal framework that is complementary to that offered by reinforcement learning alone. For this reason, it is worthwhile highlighting how the Free Energy model relates to the conceptual model proposed herein. Whilst Friston has gone as far as questioning the need for reinforcement learning as an explanatory model of behaviour (Friston et al. [2009]), it is probably fair to say that both perspectives have something to offer; reinforcement learning for the simplicity of its models, and Free Energy for its systemic scope.

7.7.1 Reinforcement Learning

Reinforcement learning has at its roots the concept of a graph where the vertices of the graph correspond to states, and the edges of the graph relate to the possible transitions from one state to another. An agent traversing the edges of the graph incurs costs (usually associated with a specific edge) and receives rewards (usually associated with arrival at a state). The decisions that determine a path through the graph are the result of a *policy*. A policy is a mapping from the perceived state of the agent to actions to be taken when in that state. In the general case the action corresponds to the selection of a path amongst a set of possible path choices. A path through the graph corresponds to a set of edges. An optimal policy corresponds to one which ensures that the maximum rewards are acquired (for reasons of analytical tractability this concept if often phrased in terms of the minimization of costs, as in this case the optimal cost is always zero). Initial work carried out on solving this class of problems was conducted by Bellman (Bellman [1957]) leading to the field of dynamic programming. More recently reinforcement learning (Sutton and Barto [1998]), has emerged, offering alternative methods and algorithms supporting the solution of such problems.

In addition to the *policy*, there are three additional elements of a reinforcement learning system; a *reward function*, a *value function* and optionally a *model* of the world or environment. The reward function maps a single state (or state-action pair) to a single value. The reward function implies the goal of the agent as the reinforcement learning system will seek to maximize long-term rewards. The reward value reflects the immediate benefit that is gained or acquired at the current state of the agent. In contrast, the value function reflects the long-term benefit of a state (or state-action pair). The value function factors in not just the immediate benefit, but also the long-term benefit that can be acquired or gained from the transition to a specific state or the choice of a specific action in a given state. Whilst entering a room might provide us with a *reward* of gold, the *value* of entering the room is low if we know that exiting it will trigger a bomb explosion. The policy decision takes as its input the result of a value function evaluation, which in turn seeks to incorporate the results of current and future rewards.

Reinforcement learning problems can be solved through the use of Temporal Difference methods (Sutton and Barto [1998]) and Q-Learning methods (Watkins and Dayan [1996]). In Temporal Difference methods the value function takes into account the different reward values associated with the alternative states reachable from the current state. In Q-Learning the value function uses reward values assigned to a specific state-action pairs. The solution of reinforcement learning problems using these methods relies upon *state value function policies*, and *action value function policies* respectively. The *state value function policy* seeks to assign a value to a state. Solutions that make use of this, do so by selecting an edge (corresponding to an action) that ensures that the next state has the highest value amongst the alternative states. In contrast, the *action-value function policy* seeks to assign a value to an action (edge selection) given the current state.

Typical implementations involve the generation of a tables of values associated with specific states (Temporal Difference methods) or actions in specific states (Q-Learning methods) acquired through the random exploration of state (or action state) alternatives. Whilst exploration is initially random, the degree of randomness declines over time as reward values are accumulated. The tension between exploration and value driven state or action state choices over time is a central theme in reinforcement learning research. More sophisticated approaches support a world model internal to the agent the accuracy of which is improved over the course of agent-world interaction. The internal world model is then used by the agent in the process of evaluating the anticipated consequences of actions.

Although the concepts of reinforcement learning methods remain valid in helping us understand the model proposed in this thesis, there are few direct comparisons that can be made. reinforcement learning methods place great emphasis on the value function, the results of which underlie the decisions that comprise agent polices. Although multi-layered perceptrons are sometimes used, for example in mapping sensor state variables to reward values (Bertsekas and Tsitsiklis [1996]), they are employed for the purposes of function approximation, not for any attempt at biological realism. In the Temporal Difference method, value estimates are continuously updated due to the difference between the anticipated value of a state and its experienced value.

In our proposed model adaptation in the network only takes place once a reward has been detected. The balance between the random exploration of states and value driven state selection is an emergent property of the neural models used. This contrasts directly with many reinforcement learning methods where the degree of exploration over time and very often the learning rates themselves are set to vary explicitly over the course of a single trial. Although a mapping is made between sensor state and selected actions, these are implicit in the dynamic connections made within our neural model. There is no explicit value function. As a consequence, no analogue between the relationship of the value function with the policy. The reward function is implicit in the world simulation itself, so is taken as given. Essentially what this thesis describes is a biologically inspired model of reward adaptation with very little in common with the formal methods of reinforcement learning. In the primitive neural circuits proposed, no internal model is constructed of the world.

7.7.2 The Free Energy Model - 'Prediction Error'

The theoretical biologist placed Anticipation as a fundamental concept in biology Rosen [1985]. He sought to promote the idea that all living organisms were in some sense Anticipatory Systems. Although this idea may have failed to gain a significant foothold in his lifetime, there are an increasing number of scientists who are prepared to advocate the idea that anticipation and prediction are potentially highly significant ideas for understanding the organization of the brain (Graybiel [1998], Bar [2009], Friston and Kiebel [2009]). Of these, perhaps the most ambitious framework is that proposed by Karl Friston. The presentation of his ideas, whilst somewhat daunting in its scope, is worthy of mention, given the benefits of comparing the model described herein with the more formal framework proposed by Friston.

Central to the idea of "Action Selection as Anticipation" is that, by default, we follow a chain of actions that are deemed to anticipate the acquisition of a Primary Reward. The accidental discovery of Primary rewards results in a backwards induction of sensor-action predictive mappings, which ensure that the action sequences that resulted in achieving this state are replayed whenever any

point in the action sequence is rediscovered. It is only on failed expectations that these action mappings are re-adjusted. By seeking to internalize predictive mappings that include only those action sequences that on average are expected to yield rewards, we reduce the dimensionality of the search state space.

There are two quotes taken from one of his more accessible papers (Friston [2010]) that are of interest for comparison purposes;

> **Optimizing the sufficient statistics (representations):** "Optimizing the recognition density makes it a posterior or conditional density on the causes of sensory data: this can be seen by expressing the free energy as surprise $Inp(s, |m)$ plus a Kullback-Leibler divergence between the recognition and conditional densities (encoded by the internal states in the figure). Because this difference is always positive, minimizing free energy makes the recognition density an approximate posterior probability. This means the agent implicitly infers or represents the causes of its sensory samples in a Bayes-optimal fashion. At the same time, the free energy becomes a tight bound on surprise, which is minimized through action".

> **Optimizing action:** "Acting on the environment by minimizing free energy enforces a sampling of sensory data that is consistent with the current representation. This can be seen with a second rearrangement of the free energy as a mixture of accuracy and complexity. Crucially, action can only affect accuracy (encoded by the external states in the figure). This means that the brain will reconfigure its sensory epithelia to sample inputs that are predicted by the recognition density in other words, to minimize prediction error".

Whilst the language may not be very easy to understand without considering his work in some detail, there are two ideas here that are well represented in the conceptual model presented;

- **Optimizing the sufficient statistics**: The use of the SOFM is employed as a means of ensuring the efficient storage of salient data in two regards.

The first is in the conventional use case, where high dimensional sensor data is represented in a far lower dimensional space. The second is based on the fact that we *only* latch data into the SOFM when a sensor event has contributed to a chain of events that has resulted in a Primary Reward. This allows us to have an efficient representation of salient data necessary for action.

- **Optimizing action:** The representation of *known* state data is minimized through the use of the SOFM. In turn, these *known* states are mapped to channel switching units which ensure that relevant sensor source data is routed to appropriate motor target units, in a fashion that seeks to limit activity to that which in the past has been demonstrated to be beneficial when sensor-motor pair sequences are linked together. It does this in a fashion such that failed expectations are used to modify the system, so that improved predictions can be employed in the future.

7.8 Conclusion

The model described in this chapter might to be of use as working functional model to researchers wanting to hand-code a reinforcement learning system. However, this is not our goal. Our goal in this chapter was to try and build a highly abstracted *model* of how the basal ganglia, thalamus and cerebral cortex working in conjunction to support reinforcement learning.

The approach taken here, mirrors that of the methodology demonstrated in chapter 5, where we hand-coded a finite-state machine model intended to solve the ball collection task. In chapter 5 the goal was to determine if there were intrinsic features of our hand-coded solution to the problem that might have brain homologues. By doing so, we were able to identify the basal ganglia as an area of particular interest. In a crude sense, we built a *mechanistic* model and then used this as a working hypothesis of the functional anatomy required to solve this class of problems. In this chapter, we are doing the inverse. We are taking insights from biology (based on our review of the basal ganglia described in chapter 6), and seeking to determine how we might implement characteristics

of these systems in a *mechanistic* model.

We stop short of coding this *mechanistic* solution however, since our goal is to arrive at a *analogue* solution that is tractable to evolutionary search. In the process of devising such a model as described in this chapter, we are creating an abstraction that represents a potential *target* for evolutionary search. In so doing we are tracing out the degrees of freedom that a more primitive circuit has the potential to explore through evolutionary search. To implement the model as described leads us to a circuit form that is intractable to evolution. However, a simpler model (incorporating some of the characteristics of this target), could be evolved without the inherent limitations of a hand-coded solution. The basis for such a solution (again inspired by biological systems) is explored in the proceeding chapter.

Chapter 8

Energy, A Neuron Model and Simple Circuits

8.1 Introduction

Whilst the previous chapter took a 'top-down' perspective, this chapter takes a 'bottom-up' perspective. We focus on one particular aspect of dopamine signalling as a starting point; the depression of phasic activity after the circuit has become conditioned to expect a reward, but the reward is not presented (see figure 8.2), in other words a failed expectation.

We take the unconventional route of considering what role, if any, energy efficiency might have to play in the emergence of these characteristics. Following this, we present a novel neuron model, which through the modulation of excitability, is able to represent some of the observed characteristics of dopamine neurons. We then present a minimalist circuit, that might present us with a good starting point in building reward anticipatory circuits (whilst still broadly complying with the framework described in the previous chapter summarized in the schematic shown in figure 8.1).

Before we go into the details, we will first present a brief review of current ideas on the role of energy efficiency in the brain.

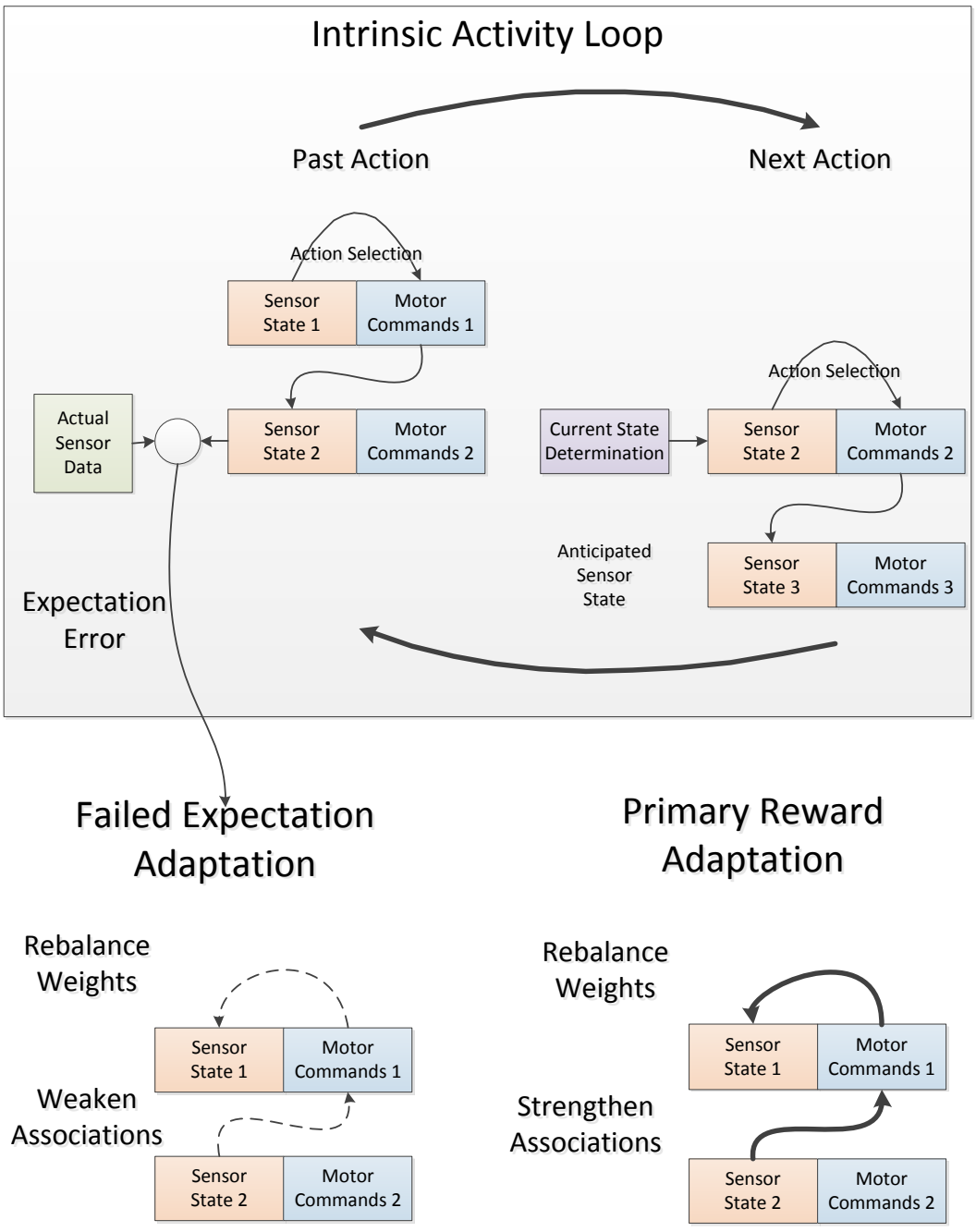


Figure 8.1: **Intrinsic Action Loop:** Unless we experience a failed expectation, action selection is automatic, based on past experience.

8.2 Energy and the Brain

It is self-evident that all replicators require energy and materials to maintain their integrity and reproduce. The efficiency with which an organism acquires and makes use of the resources that it ingests will therefore have an impact on its preferential selection (in an evolutionary system). Whilst one might argue, for example, that success in hunting is a key determinant of fitness, and therefore far more important than the energy expended in achieving hunting success, it does not follow from this that energy efficiency will not be selected for.

A simplistic understanding of evolution might lead us to see adaptations as either functional or energetic. The reality is that evolution doesn't care. If a particular mutation is functionally beneficial *and* at the same time, it improves energy efficiency, we are not able to assign exclusive credit to either 'cause', when this mutation is preferentially selected. An important point to note here, is that in identifying a particular characteristic of neural signalling, a good functional understanding of this characteristic, does not preclude an alternative explanation from the perspective of energetics. In this sense, there may well be numerous examples of dual explanations of neural activities, that are neither redundant nor mutually exclusive. This is particularly relevant to us in this thesis, as the biological role of dopamine is well researched, but its functional significance is not understood in any detail other than that it might be involved in reinforcement learning. An energetic explanation for the signalling dynamics of dopamine does not in any way subtract from interpretations of its functional significance.

In animals with the same hunting abilities, all that is required for evolution to preferentially select one set of alleles over another is *any* difference between the fitnesses of the phenotypes that correspond to each set of alleles. For this reason it is not surprising that energy efficiency is seen to play a role in; ecologicial organization (Jorgensen and Bendoricchio [2001]), cell dynamics (Haynie [2001]), ethology (Gates [2003]) and biomechanics (Alexander [2003]).

It seems reasonable to assume that energy efficiency is an important factor in the evolution of the brain. The brain is regarded as highly efficient in its support of neural signalling by some authors (see Laughlin and Sejnowski [2003] for a good overview, with support from Achard and Bullmore [2007]). Note that this does

not mean that energy utilization is minimized, as additional energy expenditure is incurred to offset signalling errors (Levy and Baxter [1996], de Polavieja [2002]). Therefore energy efficiency in signalling is the trait that is selected for, rather than simply energy minimization. Coding regimes are not uniform across all brain regions, as the functional requirements of the brain also shape the cost benefits of one signalling regime over another (Kreiman [2004]). However, there is evidence that synaptic efficiency is key in retinal processing (Vincent [2003]). Evidence also supports the hypothesis that; ion channels in a wide variety of neurons minimize energy expenditure in their normal range of spiking (Hasenstaub et al. [2010]); that dendritic processing is energy efficient (Migliore and Culotta [1998]) and that energy efficiency is a factor in the organization of vision (Vincent et al. [2005]) and other sensor systems (Niven and Laughlin [2008]).

One might argue that energy efficiency is seen not only at the micro scale in the brain, but also the macro scale through the phenomena that are said to exhibit *repetition suppression* (Grillspector et al. [2006], Schacter et al. [2007], Suzuki and Johnson [2008], Horner [2008], Constantino et al. [2008], Garrido et al. [2009], and León-Carrión et al. [2010]). Repetitive suppression refers to the decline in neural activity that takes place over the course of repetitive cognitive tasks. Whilst this may well be an emergent property of multiple low-level homeostatic processes, the results appear to be energy conservative, and as such, may play a role in contributing to overall fitness.

The energetic costs of single neuron spike is high. There are estimates that this may limit the human cortex to fewer than 1 percent of the number of neurons that can be substantially active at the same time (Lennie [2003]). Rather than an abundance of signalling, the lack of activity in the brain has given rise to the notion of the 'dark matter' of neuroscience (Shoham et al. [2006]), since we have difficulty in accounting for such low levels of signalling. It could be argued that low activation levels are consistent with the idea of an energy efficient brain.

An argument that is sometimes presented in support of the idea that the brain is *not* energetically efficient, and therefore unlikely to have been the target of selection for energy efficiency, is that the energy used by the brain at rest is not significantly different from that of the brain at full use. This is a specious argument. If a computer consumes close to the same amount of energy on standby

as it does in normal operation, it could either be incredibly efficient or incredibly inefficient. Energy efficiency cannot be determined without determining a basal metabolic rate of energy expenditure and using this as our reference rate. Reports that suggest that the brain uses approximately 50 percent more energy when conscious compared to when it is under deep anaesthesia (Laughlin and Sejnowski [2003]), still does not give us enough information to say whether or not this is 'high' or 'low', since we have no basis of comparison. If the base metabolic cost is minimized, then knowing that the brain uses 50 percent more energy when conscious than when unconscious does not imply that the brain is inefficient.

If we were to find a popular car, with a very small petrol engine we might imagine it to be an indicator that the car was energy efficient. Analogously, given that the brain has very little energy storage capacity (Ames [2000]), we might arrive at the same conclusion. What has been established is that peak energy expenditure is not substantially different from that of its mean (Raichle and Snyder [2007]). Whilst it could be claimed that this mean is energy inefficient (which is unlikely), the conclusion that we have to come to is that the functional activity that we associate with cognitive tasks, does not result in significant increases in energy expenditure, surely an indicator that the neural signalling that supports these cognitive tasks *is* energy efficient.

We know that energy consumption of the brain is significant when considering total human energy expenditure at rest. Although the human brain is only 2 percent of the body's weight, it accounts for 20 percent of its resting metabolism (Attwell and Laughlin [2001], Laughlin [2001]). Saris (Saris et al. [2008]) estimates that in humans this may be as high as 20-25 percent. This is substantially higher than that for other primates (8-10 percent) and non-primate mammals (at 5 percent). It is unlikely that this relatively high metabolic cost would not be subject to genetic adaptation through preferential selection. Some authors (Laughlin et al. [2000] and Hasenstaub et al. [2010]) go as far as to propose energetics as a single unifying principle in understanding brain signalling.

If we accept that evolution has played a role in improving energy efficient, we might ask ourselves whether or not this is implied by any general characteristics of neuronal signalling. Perhaps the most obvious area to consider is that of neural inhibition. We know for example that cortical circuits rely upon GABAergic inhi-

bition to balance excitation and control spike timing (Mann and Paulsen [2007]). Buzsáki et al (Buzsáki et al. [2007]) have also identified the role of inhibitory interneurons as key in supporting energy efficiency. The ideas expounded in the rest of this chapter, whilst originally inspired by certain dynamics of the dopamine signalling, apply more generally to that of the action of inhibitory neurons, and their role in supporting energy efficiency.

8.2.1 Dopamine Signalling and Energy Regulation

There is one significant feature of dopamine signalling that is potentially indicative of neural adaptation that supports energy efficiency (see figure 8.2); after training, a failed reward expectation results in the inhibition of dopamine phasic activity below basal levels (Schultz [1998]).

Our hypothesis, portrayed in figure 8.3, is that during training we see a decline in the intrinsic excitability of the dopamine neuron. However, this decline is compensated by a corresponding increase in susceptibility or responsiveness (to the activation of the efferent dopaminergic neuron) at the afferent target neuron. Such adaptations would help explain the decline in dopamine output below background levels, when a failed expectation presumably results in an 'off' input signal to the dopamine neuron. If this model is reasonable, we also have to consider the possible regulatory pathways through which this adaptation might take pace. Three broad alternatives are portrayed in figure 8.4 which we term; (a) *auto-regulation*, (b) *modulatory regulation* and (c) *feedback regulation*. Of the three (c) appears a plausible explanation in the case of dopamine neurons, since striatal neurons make inhibitory projections to the substantia nigra (the origin of the dopminergic neurons). In this scenario, the output of the target neuron feeds back to the innervating dopamine neuron, to down-regulate dopamine excitation, and at the same time it increases its sensitivity to the input signal, to maintain the modulatory impact of a reduced dopamine signal.

8.2.2 Intrinsic Versus Task-Evoked Brain Perspectives

One aspect of these hypothesis that is worth mentioning, as it relates to dopamine phasic activity as portrayed in the schematic 8.3, is that it relies on four distinct

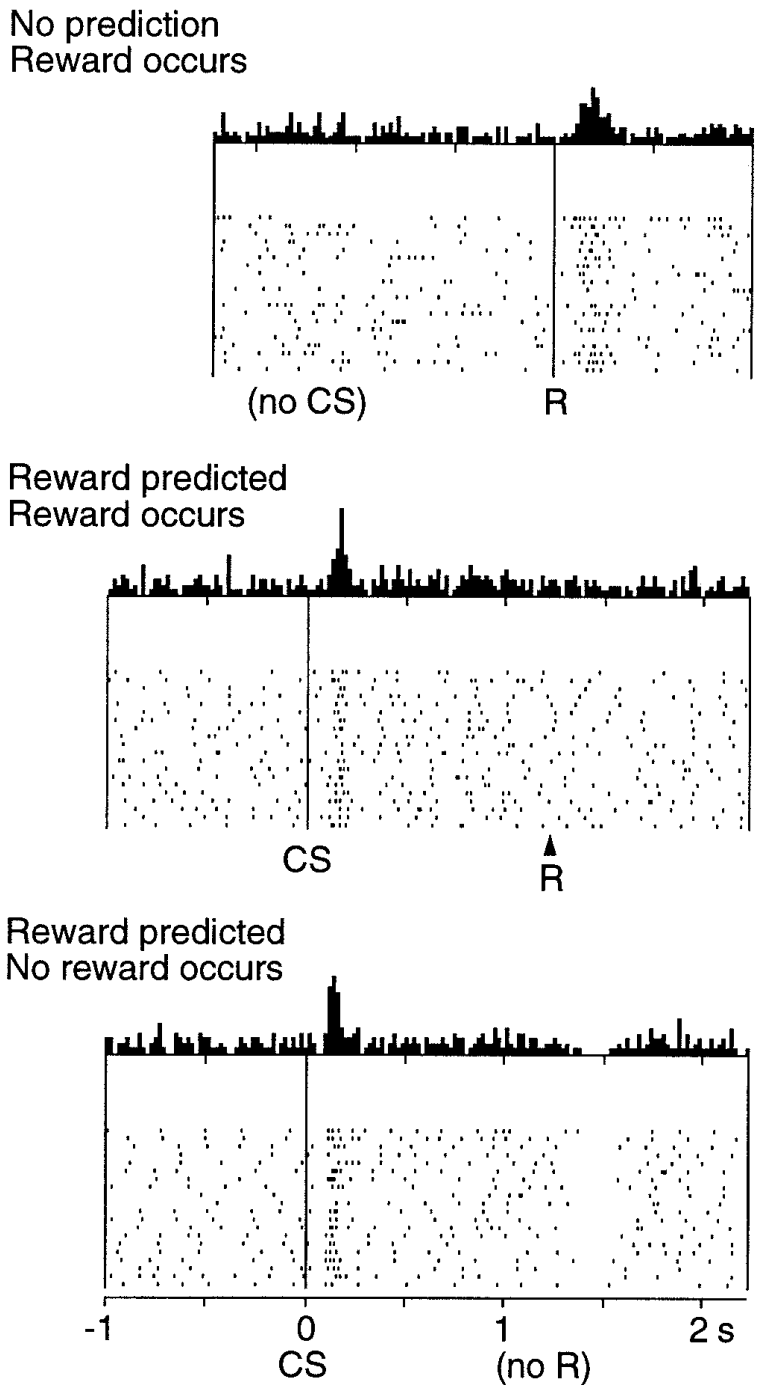


Figure 8.2: **dopamine phasic activity inhibited on a failure in an anticipated reward.** The charts correspond to time binned histograms of dopamine phasic activity in three seperate experiments. "R" corresponds to the presentation of a reward. "CS" corresponds to the presentation of a conditioned stimulus. Note that in the third diagram, the average *activation level is lower on the failed presentation of a reward* after training. Source Schultz [1998].

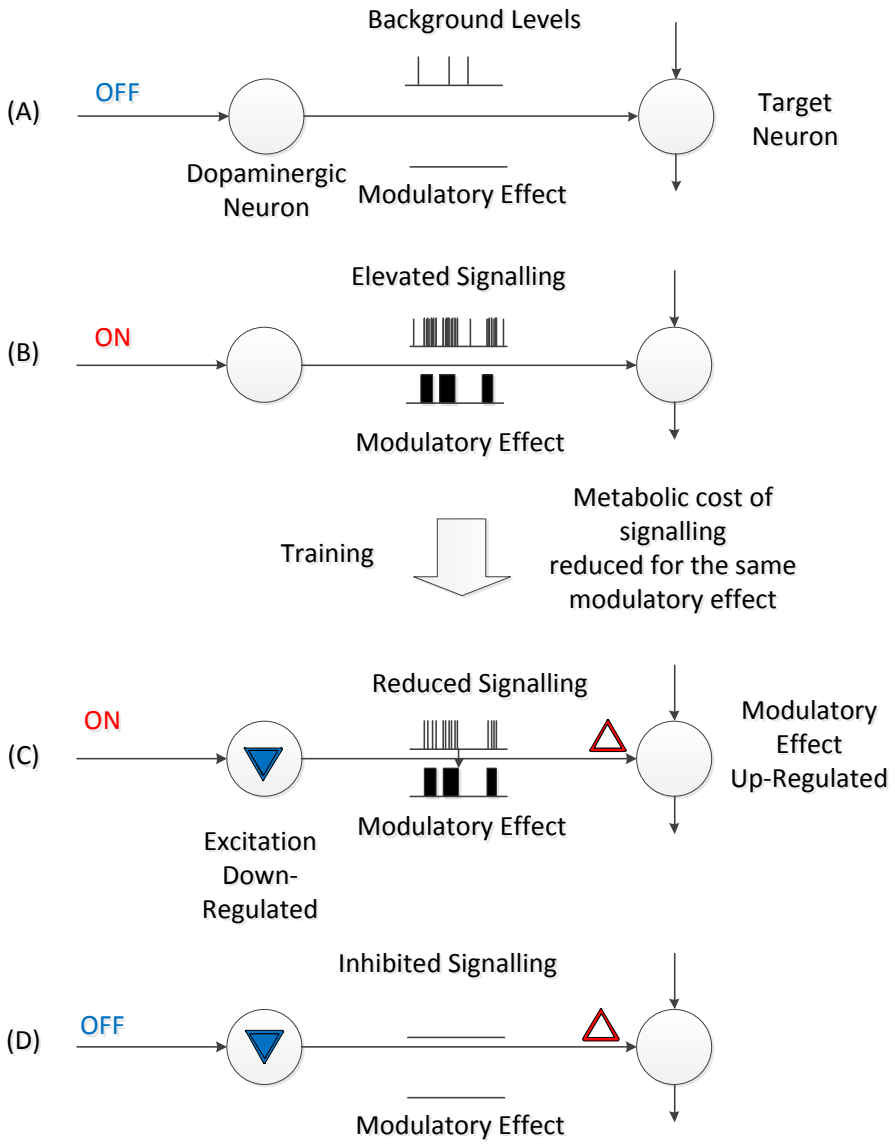


Figure 8.3: **Dopamine signalling energy regulation hypothesis: (A)** Depicts a dopamine neuron with a target neuron. When it is 'off', the dopamine neuron exhibits signalling at background levels of activity. **(B)** When the input is 'on' we have elevated signalling levels. **(C)** During training the excitability of the dopamine neuron is gradually reduced, and the receptivity of the target neuron is up-regulated. **(D)** Subsequent to training, when dopamine input is once again 'off' (e.g. denoting a failed expectation), source output levels are substantially inhibited, below the background levels that were seen prior to training. The downwards-pointing (filled, blue) triangle: denotes reduced excitation. The upwards-pointing (empty, red) triangle: denotes enhanced receptivity.

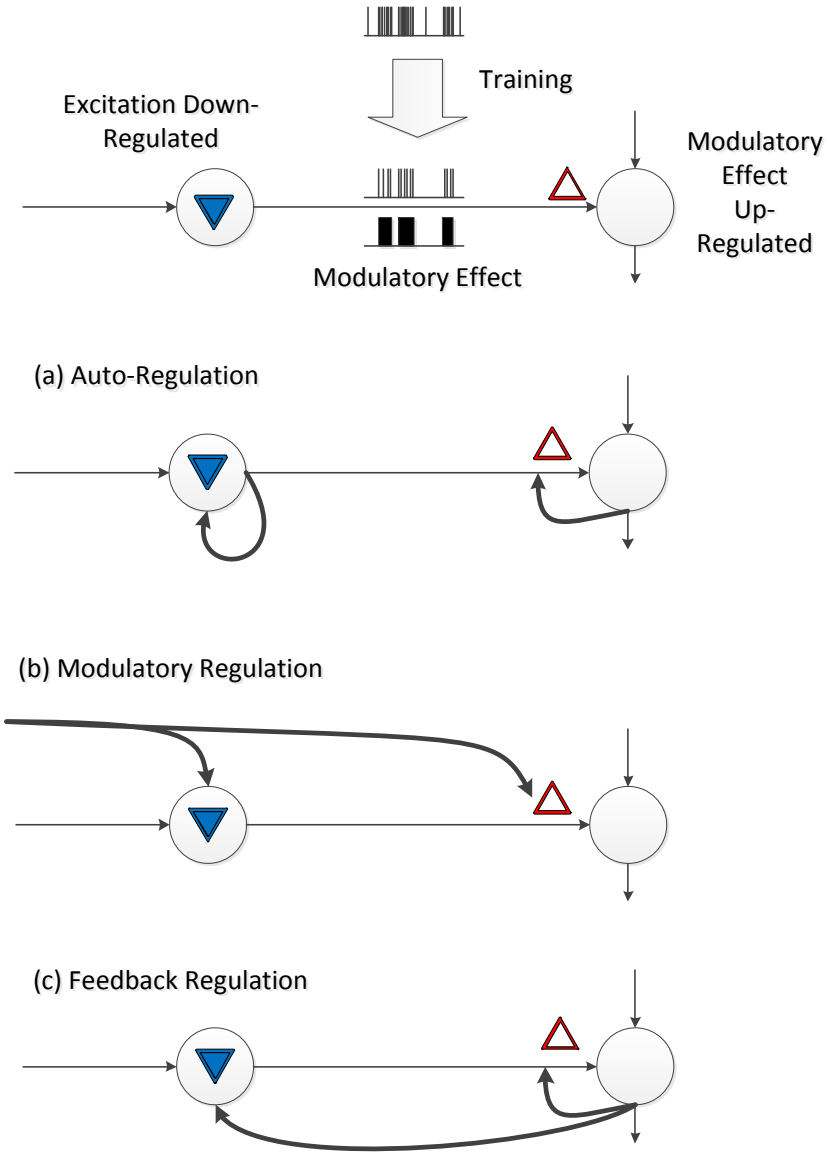


Figure 8.4: **Energy Regulation Hypothesis:** Three possible models that may account for the changes that appear to take place in dopamine activation levels during training. **(a) Auto-Regulation;** dopamine excitation down-regulation and target neuron receptivity up-regulation are assumed to be self-regulating due to some homeostatic regulation and hebbian adaptation respectively. **(b) Modulatory Regulation;** an external source provides a modulatory signal to both source and target neurons. **(c) Feedback regulation;** in this the target neuron, down-regulates the dopamine (source) signal, and a hebbian process adjusts the target neurons susceptibility to this source signal. The downwards-pointing (filled, blue) triangle: denotes reduced excitation. The upwards-pointing (empty, red) triangle: denotes enhanced receptivity.

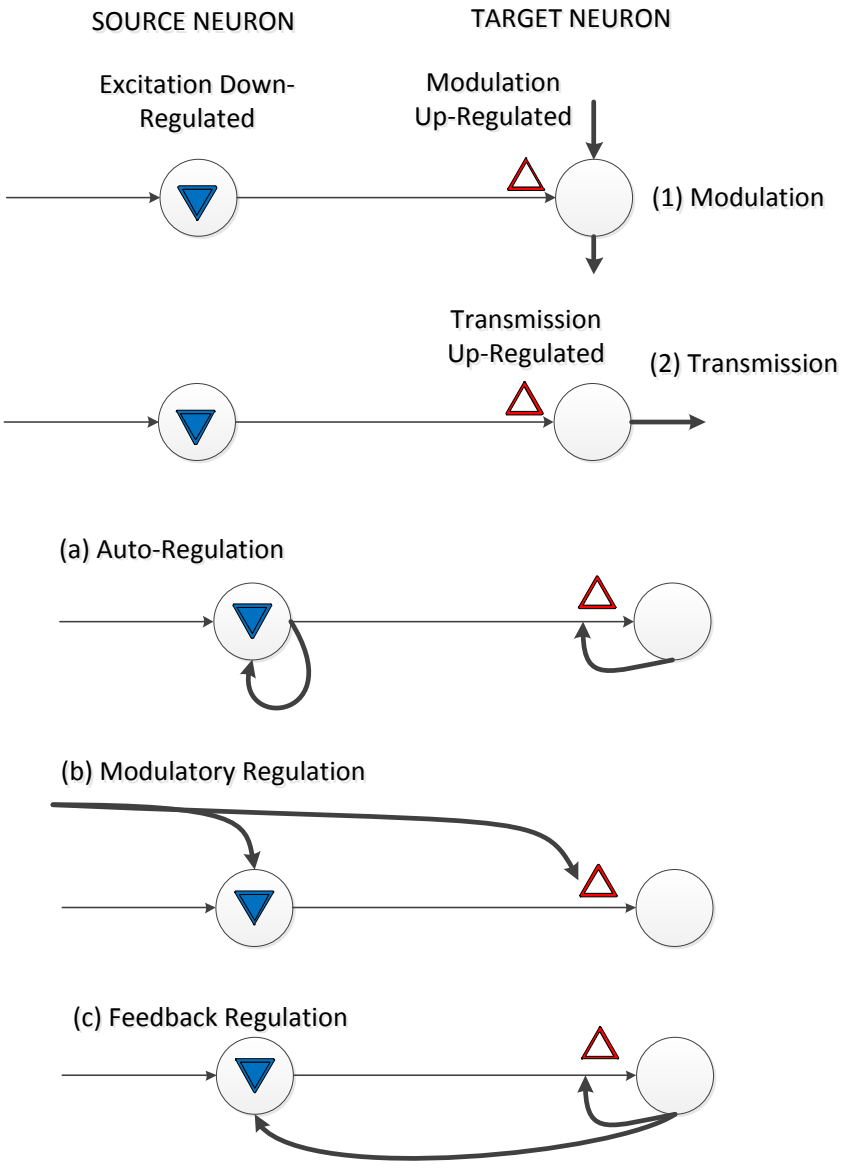


Figure 8.5: **Generic neuron energy regulation hypothesis:** Whilst 8.4 indicates the different possible mechanisms for energy regulation for use in the modulation of the target of a dopaminergic neuron, any generic *source neuron* might regulate a *target neuron*, either for purposes of modulation or transmission via (a) auto-regulation, (b) modulatory regulation or (c) feedback regulation. The downwards-pointing (filled, blue) triangle: denotes reduced excitation. The upwards-pointing (empty, red) triangle: denotes enhanced receptivity.

signalling patterns; (1) background, (2) elevated, (3) reduced elevation and (4) inhibition. However, there is little support in literature for a distinct pattern that corresponding to reduced elevation. We may be able to explain this by suggesting, that after a process of adaptation, the dopamine signalling, tends to the background level. The consequence of this, would be that dopamine neurons that have been trained, are indistinguishable from other dopamine neurons that have not been trained, if we were to look at their signal characteristics. In this case an 'adapted' circuit, is not detectable by its signalling levels. It would only be discernible by considering the excitation state of the *source* (dopamine in this case) neuron, and the susceptibility or input weight of the pre-synaptic neuron at the *target* neuron.

The conceptual implications of this are potentially far-reaching, as this implies that inhibitory neurons are potentially engaged in activity that is intended to down-regulate source neurons, when it is known that these neurons are likely to be involved in significant signalling (offset by in up-regulation in target neuron susceptibility). In this sense, inhibition denotes the importance of the signal that originates from the *source* neuron. This changes substantially what we might presume is a natural interpretation; that inhibitory neurons are "stopping" signalling that is unimportant. No doubt there are cases (particularly in interneurons) where we see examples of competitive inhibition. However for non-local inhibitory projects, it could potentially be the case that the functional role of inhibitory neurons is to regulate the output of principal contributory *source* neurons, quite possibly in the case of the Feedback Regulation model, by the very *target* neurons that they innervate.

If key signalling is mediated by coupled neurons that engage in commensurate source neuron (excitability down-regulation) and target (pre-synaptic strengthening) adaptation, then the energy footprint of active signalling would be reduced. Such signalling would not necessarily be significantly greater in energy expenditure than that of unadapted neurons at their rest metabolic state. Such adaptation is possible where signalling is intermittent, but in some sense predictable. Bursts of elevated activity would still be seen when irregular or unanticipated signals occurred. Such signals might appear to correspond to novelty, or surprisal (Palm [1981], Strange et al. [2005], Friston [2010]).

Raichle ((Raichle and Gusnard [2002]), Raichle [2006]) differentiates between *intrinsic* and *task-evoked* signalling. His view is that much of our understanding of brain functionality is dependent on experimental methods that evoke neural signalling as part of a task. He argues that this bias might significantly distort our understanding of the brain, since it neglects what he terms intrinsic brain activity.

Raichle does not specify what this intrinsic activity might be. A process of active energy management (as outlined above), might seek to minimize high-traffic connections through paired source and target neuron adaptation. Such signalling would be functionally significant, but it not be highly visible.

8.3 The Neuron Model

8.3.1 Desired Characteristics

The following model is intended to capture certain features of the dopamine neuron, specifically; an excitation factor that we can adjust to modify the firing probability of the neuron. Characteristics that we require in such a model include;

1. The probability of the neuron firing should increase monotonically with an increase in the neuron's internal activation state.
2. The neuron should still fire stochastically even with neutral input.
3. We require an excitation parameter that allows us to adjust the probability of firing.
4. It should be computationally simple to implement.
5. Ideally the formulation should support both spiking models and average rate code models.

8.3.2 The Rayleigh Distribution

With these characteristics in mind, we arrived at a model that uses the Rayleigh Distribution. The Rayleigh Distribution is a two-dimensional analogue of the more widely known Maxwell Distribution. Imagine a square pool table, where all collisions are perfectly elastic (i.e. energy is conserved), and all balls are given an initial random velocity (whose horizontal and vertical components are uncorrelated, generated from a normal distribution with the same variance). If we then took the speed of each ball, we would find that the speeds corresponded to a Rayleigh Distribution. If the pool table was mounted on a motorized platform, we would be able to agitate the table, thus adding energy to the system, resulting in an increased in the average speed of the balls, analogous to increasing the 'temperature' of the system.

The Rayleigh Distribution has a Probability Density function given by;

$$\frac{x}{\lambda^2} e^{-z^2/2\lambda^2} \tag{8.1}$$

and a Cumulative Distribution Function given by;

$$1 - e^{-z^2/2\lambda^2} \tag{8.2}$$

Where z denotes the random variate and λ corresponds to a "temperature" parameter, that determines the shape of the distribution.

8.3.3 Average Rate Formulation

In this section we describe the implementation details of our neuron model. We arrive at a formulation based on the Rayleigh Distribution, which gives us a neuron firing probability determined by a neuron **activation level** (or **State**) and an **excitation factor**. Non-spiking neuron models of neurons typically have a single output value that is intended to correspond to an average firing rate. A neuron with a fixed probability of firing for a given activation level and excitation factor would a generate stream of pulses whose average firing rate is directly analogous with that of other average rate firing models. The biggest difference between our formulation and that of other commonly used average rate firing models is that the peak and base output values vary with excitation factor, together with the shape firing probability curve. In GasNet and other adaptive network models (described in Chapters 2 to 4), whilst the shape of the curve can be modulated, the peak and base output levels are always mapped to a interval either 0 to 1, or equivalently -1 to +1. This results in a model whose characteristics bear a closer resemblance to biological neurons than other average rate code models, but without involving the complexity and computational overhead of some spiking neuron models.

In our neuron model, the probability of the neuron firing (p) is determined by the Rayleigh Distribution's cumulative probability function;

$$p = 1 - e^{\frac{-z^2}{2\lambda^2}} \tag{8.3}$$

We define two new variables x and y, which correspond to our neuron **excitation factor**, and **activation state** respectively. With an appropriate mapping, the activation state (y) determines the λ of the distribution, such that an increase

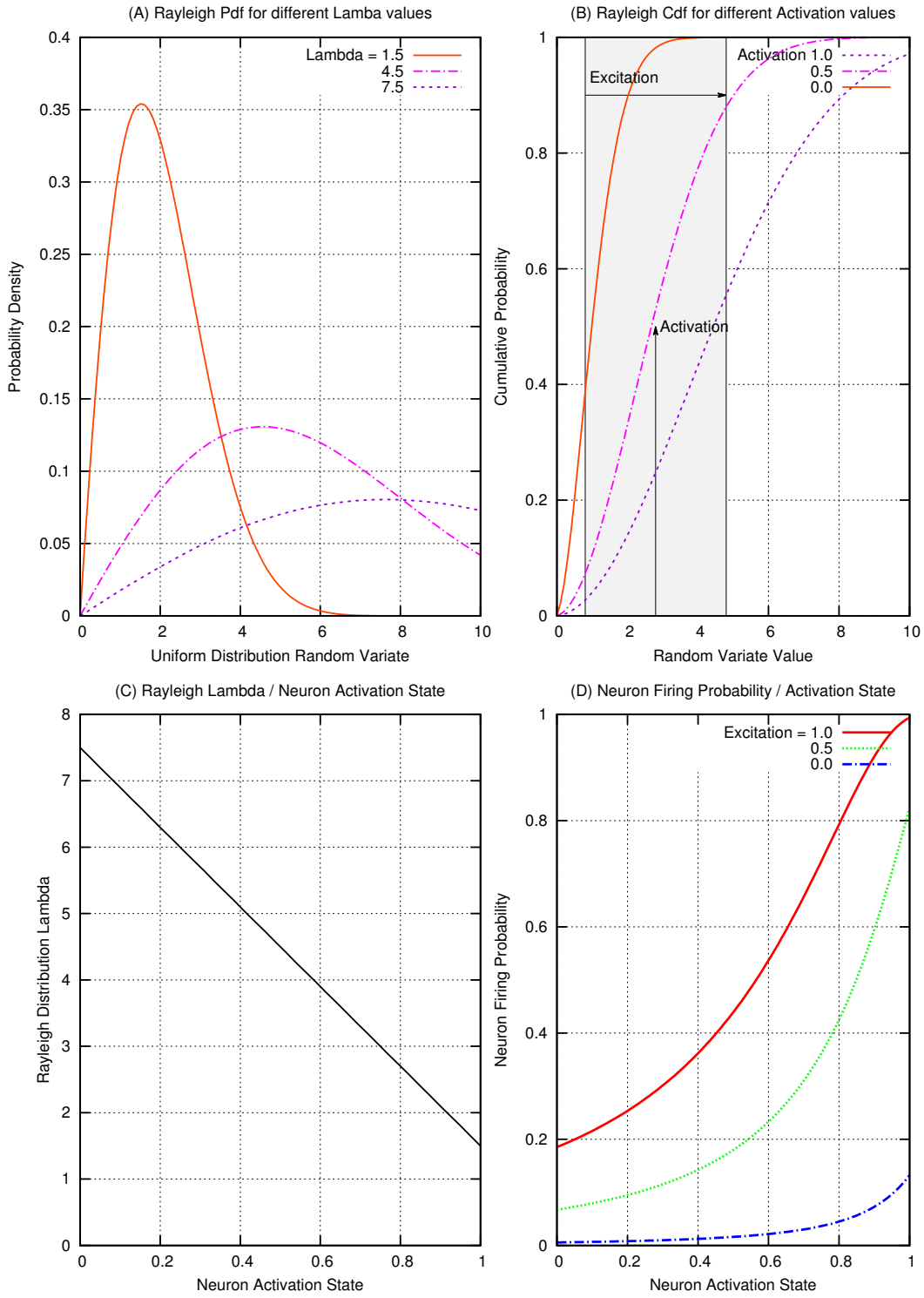


Figure 8.6: **Neuron Model: (C)** We map the neuron activation state to a Rayleigh distribution Lamda value **(A)**. The excitation factor acts a threshold based on the cumulative probability **(B)**. This gives us a Neuron firing probability for an activation state and excitation factor **(D)**.

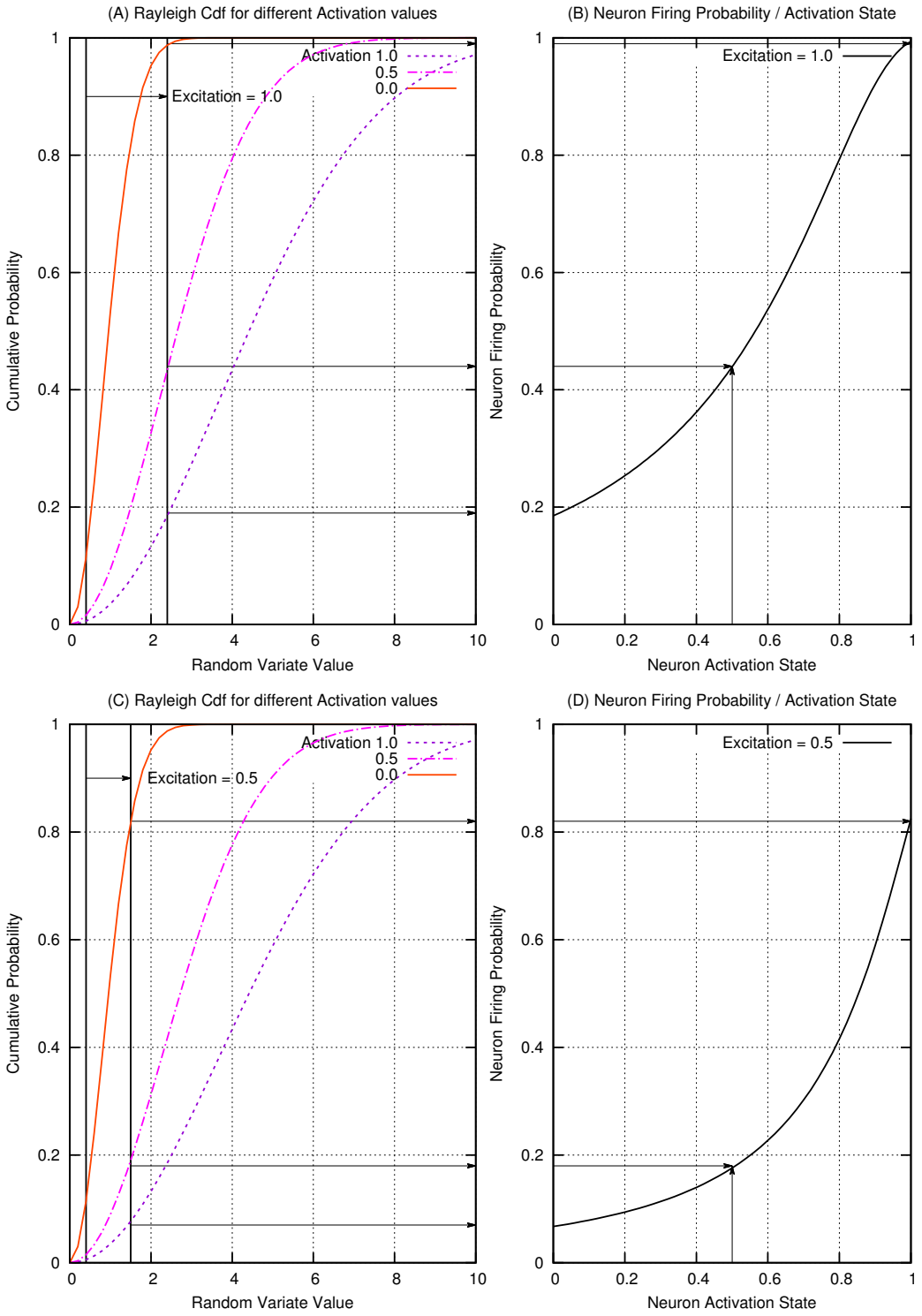


Figure 8.7: **Neuron Firing Probability: (A)** Firing Threshold set by excitation factor at **1.0**. **(B)** Firing Probability at excitation factor at **1.0**. **(B)** Firing Threshold set by excitation factor at **0.5**. **(D)** Firing Probability at excitation factor at **0.5**.

in the activation state corresponds to a decline in the value of λ, and vice-versa. Qualitatively, we might imagine the activation state to be inversely related to the 'temperature' of the Rayleigh Distribution. The excitation factor x acts as a threshold value, and is mapped to the z variate of the Rayleigh Distribution. Where x corresponds to the neuron's **excitation factor** and z the cumulative probability variate in the Rayleigh Distribution;

$$z = (k_1 + k_2 x) \tag{8.4}$$

and where y, corresponds to the neuron's **activation state** and λ the "temperature" parameter of our Rayleigh Distribution;

$$\lambda = (k_3(1 - y) + k_4) \tag{8.5}$$

we obtain the neuron firing probabability (p) given by;

$$p = 1 - e^{\frac{-(k_1 + k_2 x)^2}{2*(k_3(1-y)+k_4)^2}} \tag{8.6}$$

Neurons exhibit a wide range of firing probabilities (see Branco and Staras [2009] for a review).The constant values chosen for equation8.6 were intended to reflect this. Striatal Medium Spiny Neurons are estimated to fire with a probability between 0.42 and 0.72 (Ding et al. [2008]). In our model with the excitation factor at 0.5 (intended to be a neutral level), the neuron will fire with a probability of between 0.2 and 0.8 (with activation levels at 0.5 and 1.0 respectively.

For all our experiments, the constant values are taken to be; k_1 (0.4), k_2 (2.0), k_3 (6.0) and k_4 (1.5). The value for k_1 (0.4) was chosen to that all neurons fire with a small but non-zero probability even when the excitation factor (x) is zero, and the activation state (y) is zero. The value for k_2 (2.0) was chosen, to ensure that the relatively small changes in excitation factor (x) will result in quite substantial changes in the form of the neuron firing probability for the ranges λ, determined by the activation state (y) variable. The values relating to activation state (y) k_3 (6.0) and k_4 (1.5) were chosen in conjunction with those for the excitation factor

(x), to ensure firstly that; the firing probability (p) was close to 1.0, when the neuron has an activation state of 1.0, and an excitation factor of 1.0 and secondly that the firing probability (p) was not too high, when the activation state is 0.0, with an excitation factor of 1.0.

This model is displayed graphically in figure 8.6. In figure 8.6, graph A shows the Rayleigh Probability Density function for different λ values. For any given Neuron activation level, we can determine the appropriate λ value to use, as shown in graph B (in line with equation 8.5). Once we have the appropriate λ value, we can use this to determine the appropriate the curve for the Rayleigh distribution cumulative probability function (graph C). The value of the excitation factor determines the point on the Cumulative Probability curve that corresponds to the current neuron firing probability (p), as shown in graph D.

The correspondence between the Rayleigh Distribution Cumulative Probability Function and the Neuron Firing Probability (p) can be seen even more clearly in figure 8.7. The upper two graphs show how the neuron firing probability curve is generated when the excitation factor has the value 1.0. The intercept of the cumulative probability curve's (which correspond to different neuron activation levels) with the threshold determined by the excitation factor is shown for a neuron activation of 1.0. 0.5 and 0.0. The horizontal arrows highlight these intercepts (graph A). In turn, we can see the corresponding points on the neuron firing probability curve in graph B. Similarly the lower two graphs show the same cumulative probability curves, with a new threshold determined by the excitation factor set to 0.5. (graphs C and D).

The graph shown below in figure 8.8 illustrates the neuron firing probability curves in more detail (firing probability p as a function of activation state). Let us consider the case where the excitation parameter is 0.5 (the neutral excitation level), the centre curve with a solid line. We can see that the neuron will fire with a probability of around 83 percent when the internal activation state is 1.0, and with a 7 percent probability when the activation state is zero. A reduction of the excitation parameter to zero results in a very low level of neuron firing probability. A high level of excitation results in guaranteed firing when the activation state is 1.0. The graph shows results taken from the implemented model. Figure 8.9 shows the neuron firing probability as a heat map. In this case we can see how the

variation of both excitation factor and neuron activation level affect the neuron firing probability across a range of values.

In the experiments that are described in subsequent chapters, we add the small perturbation so that neurons with the same Activation Value and excitation factor never have exactly the same output values. This is useful where we want to implement a winner-takes-all action selection algorithm, and we wish to see stochastic outcomes even with nominally identical inputs. The final form of the neuron firing probability is shown below, where δ corresponds to a very small random value;

$$p = 1 - e^{\frac{-(k_1+k_2x)^2}{2(k_3(1-y)+k_4)^2}} + \delta \tag{8.7}$$

For reasons of computational efficiency, we created a two-dimensional look-up table comprising 11 excitation factors and 41 activation levels, and employ linear interpolation for intermediate values. This look-up table (together with a small random value δ)is used in all experiments described in future chapters rather than equation 8.7 for calculating firing probabilities.

8.3.4 Excitation Modulated Signalling

By varying the excitation factor, we are able to change the neuron transfer function. The two graphs shown in 8.10 show how we might support the process described earlier this chapter, where a change in the excitation factor for a dopamine neuron results in inhibited output when the input is weak. In the Top graph, we can see transfer functions with the excitation factor of 1.0, 0.5 and 0.1 In our hypothesis regarding the dopamine neuron, we are assuming that dopamine neuron excitation declines over the course of training. If we start with an excitation factor of 1,0 which declines over time to 0.1, we can see that an internal activation state of zero, would reduce in a substantially inhibited output train with a firing probability of approximately 0.03, compared to a 'background' level of 0.18 (at point D) prior to the decline in excitation. Despite the decline in excitation, however, if the internal state of the neuron is 1.0, the neuron will fire with a probability of approximately 0.27. The lower graph shows a similar phenomena at lower activation states. We could easily imagine a system of channels,

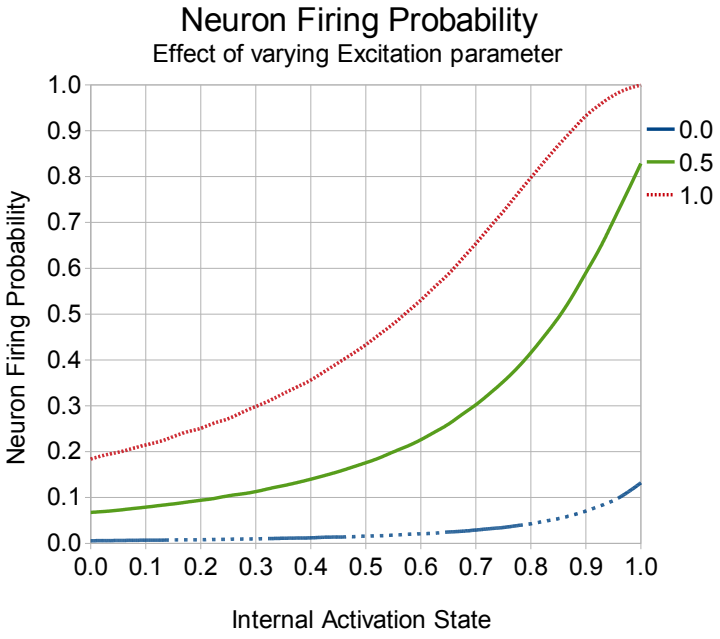


Figure 8.8: **Neuron Model- Firing Probability vs activation state:** Here we show the impact of varying the excitation factor. In both graphs above, the upper line shows the effect of the excitation factor set to 1.0, and the lowest line shows the excitation factor set to zero. Note that even when the internal activation state of the neuron is zero, the probability of firing never goes to zero.

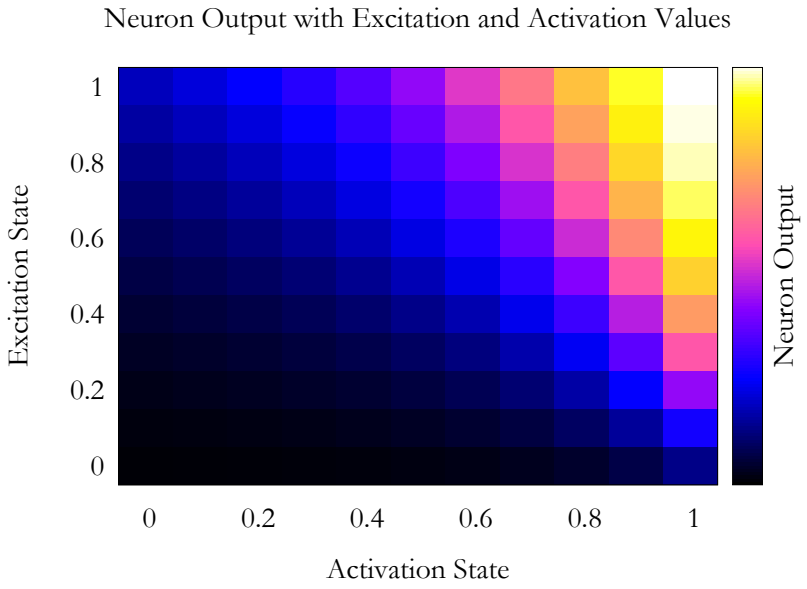


Figure 8.9: **Neuron Firing Probability vs activation state and Excitation State:** The x-axis corresponds to the activation state of the neuron, and the y axis the neuron's excitation state. The neuron's firing probability is displayed as a colour; ranging from of zero (black) to 1.0 (white).

differentiated only by their excitation factors.

8.3.5 A Spiking Formulation

It is relatively simple to generate a spiking version of this model. We can generate a random variate (z) that corresponds to the appropriate Rayleigh distribution, using the following formula;

$$z = \lambda\sqrt{-2\ln(1-U)} \tag{8.8}$$

where U is a random variate drawn from a uniform distribution in the range 0 and 1, and where λ is determined by equation 8.5 as before, based on the current neuron for an activation state y. We can then use the excitation factor as a simple threshold level, such that the neuron is said to fire if the value of z is **less** than the threshold (with appropriate scaling factors analogous to those used in equation 8.4). The implementation of this model is trivial, as shown in Listing 8.1. Note that the constants $k1$ to $k4$ correspond with those that are specified in equations 8.4 and 8.5.

Listing 8.1: Spiking Formulation of the Model Neuron

```
/// Where fActv is the neuron Activation State (Level)
/// and fExct is the neuron Excitation State.
float CModel_Neuron::Output( float fActv, float fExct  )
{
        // 0.4f (k1)
        // 2.0f (k2)
        // 6.0f (k3)
        // 1.5f (k4)
        float fLambda   = 6.0f * (1.0f - fActv) + 1.5f;
        float fThrs     = fExct * 2.0f + 0.4;
        float fRand     = Get_Rnd_Real();//(0.0,1]
        float fRayl     = fLambda * ::sqrt( -2.0f * ::log( fRand ) );
        return (fRayl < fThrs ) ? 1.0 : 0.0f;
}
```

Through the generation of a binned histogram of 5 million samples using equation 8.8, we verified that this form is equivalent to that of equation 8.7. Although we have not used this model in experiments described within this thesis, some consideration was given to how a Spike Timing Dependent Plasticity (for reviews see Dan and Poo [2004], Dan and Poo [2006], Natalia Caporale And Yang

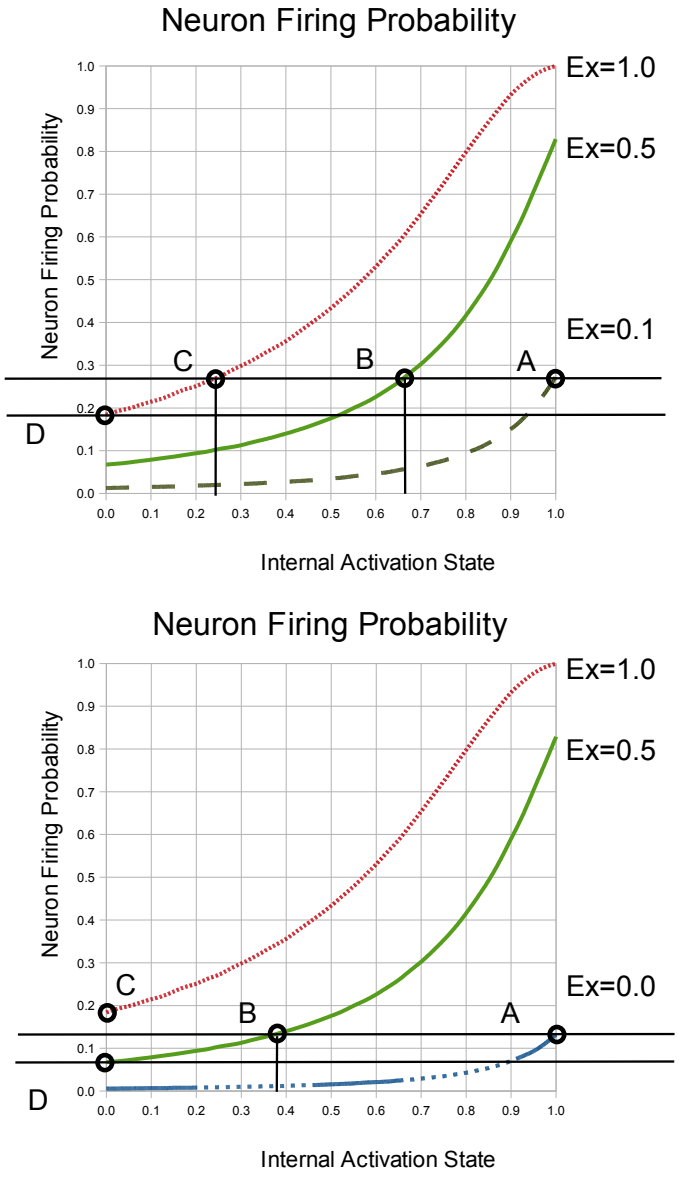


Figure 8.10: **Neuron Model - Excitation Modulated Signalling:** Here we can compare the impact of changes in the excitation factor on competitive signalling between two neurons. (Top) Even if the excitation factor of a neuron is reduced to 0.1, when its internal activation state is 1.0 (A), the probability of firing is greater than a neuron with an excitation factor of 0.5 with a 0.67 activation state, and greater than a neuron with an excitation factor of 1.0 with an activation state at 0.25. (Bottom) If a neuron has an excitation factor of 0.0, its probability of firing when its activation state is 1.0(A), will still be greater than that of a neuron with an excitation factor of 0.5, with an internal state of 0.38

Dan [2008], and for a computational perspective see Roberts and Bell [2002]) could relate to a spiking model.

There are two notable models that seek to relate Spike Timing Dependent Plasticity (STDP) to Reinforcement Learning. Roberts et al (Roberts et al. [2008]) propose a model where a STDP learning rule drives the spike probability of a reward predicting neuronal population to a stable equilibrium. The model proposed by Izhikevich (Izhikevich [2007]) is particularly interesting, since he proposes a STDP learning model where dopamine neurons provide a reinforcement signal capable of associating distal (prior) actions with a current reward. What we propose here however is something entirely different. We are interested in a simple mechanism that might support the differential regulation of connected neurons, for the purposes of supporting energetic signalling, such that the source neuron has its signal down-regulated, and the target neuron has its sensitivity up-regulated.

Work by Markham and Tsodyks (Markham and Tsodyks [1996]) showed that pairs of pre and post synaptic potentials affected synaptic efficacy, and not just synaptic output. This is supportive of the idea of gain adjustment implied by figure 8.11. Mark Denham (Denham [2001]) used this idea as the basis for developing a biologically plausible model which demonstrates that phase variations in spike arrival at pre and post synaptic terminals can result in both long-term potentiation (LTP) and long-term depression (LTD). He refers to the typical spike-timing related plasticity response illustrated in figure 8.11. Pre-synaptic signals that precede action potentials reduce synaptic efficacy, and pre-synaptic signals that occur slightly after action potentials increase synaptic efficacy. This is consistent with our model. Essentially, an anticipatory signal is theoretically capable of down-modulating *transmitter* gain, and up-modulating a *receiver* gain, resulting in energetic gains without information loss. This model qualitatively concurs with the phenomena that appears to be present in reward-anticipatory activity of dopamine neurons (Mirenowicz and Schultz [1994]). We also know that dopamine neurons are implicated in both LTP and LTD, fitting suitably with our model for a candidate anticipatory modulator. Ideally we would wish to find a biological example of where the same dopaminergic circuitry was implicated in both LTP and LTD where the candidate for LTP was subject to a phase

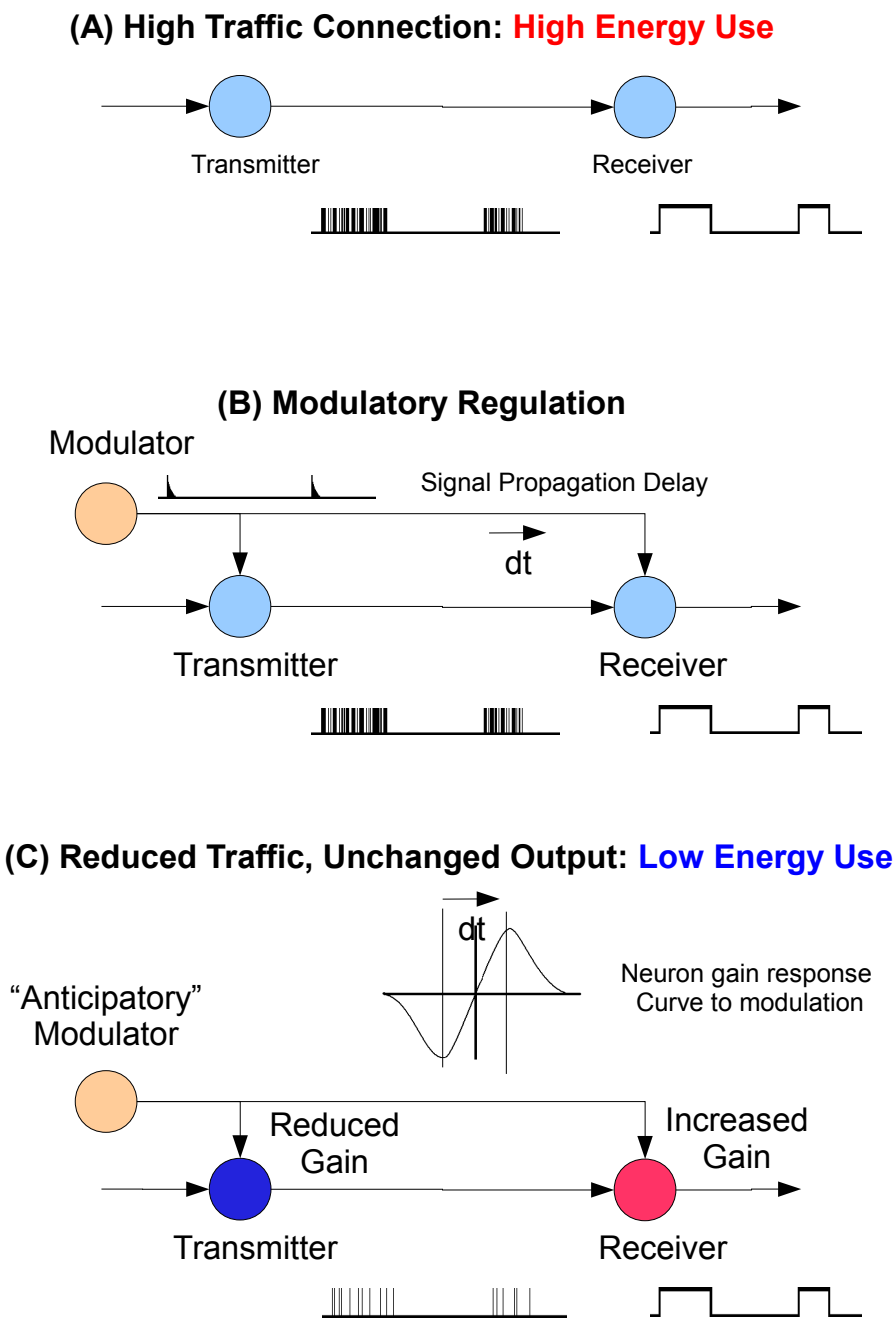


Figure 8.11: **Hypothetical STDP and Energy Efficiency:** This graph shows how a hypothetical neuron model could use where we have an Anticipatory Modulator, that through Spike Time Dependent Plasticity is able to improve Energy Efficiency of Signalling.

delay. Whilst we are not aware of such a simple circuit in current literature, we do know that striatal cells respond in a fashion that anticipates reward (Schultz et al. [2000]), consistent in our model with a modulator that is phase-delayed, with regards the original anticipatory signal.

8.4 Experiment: Energy Savings in Feature Detection

This section describes an experiment which demonstrates energy savings through a simple feature detection task. It makes use of a simple circuit comprising our model neurons. We shall refer to our model neurons as *Rayleigh neurons* implying the function used to calculate the effective output of these neurons. The circuit used in this experiment is illustrated in figure 8.12.

8.4.1 Problem: The detection of combined features

We base this test on a simple problem that imagines the existence of a organism in a simple environment dominated by two sensor domains. Domain A (e.g. taste) has 32 possible states, and Domain B (e.g. colour) has 16 possible traits. We assume to the existence of two feature detectors for each domain. A feature detector is represented by a single neuron. For the two domains, we have a total for four neurons comprising the *input-layer* to the circuit. At the start of each trial we randomly select two values from each domain and associate each feature detector with these values. At each simulation iteration we generate a random value for each state that is deemed to correspond to the current environmental state. When the environmental value corresponds to that of associated with a feature detector, we set the corresponding neuron activation state to a value of one. Otherwise the activation states of each neuron are set to zero.

In addition, at the start of each trial, we randomly select a reward state, that corresponds to one specific combination of feature detection outputs. When the feature detectors indicate that the environment state corresponds to this particular combination of values (one from each domain), we assume the existence

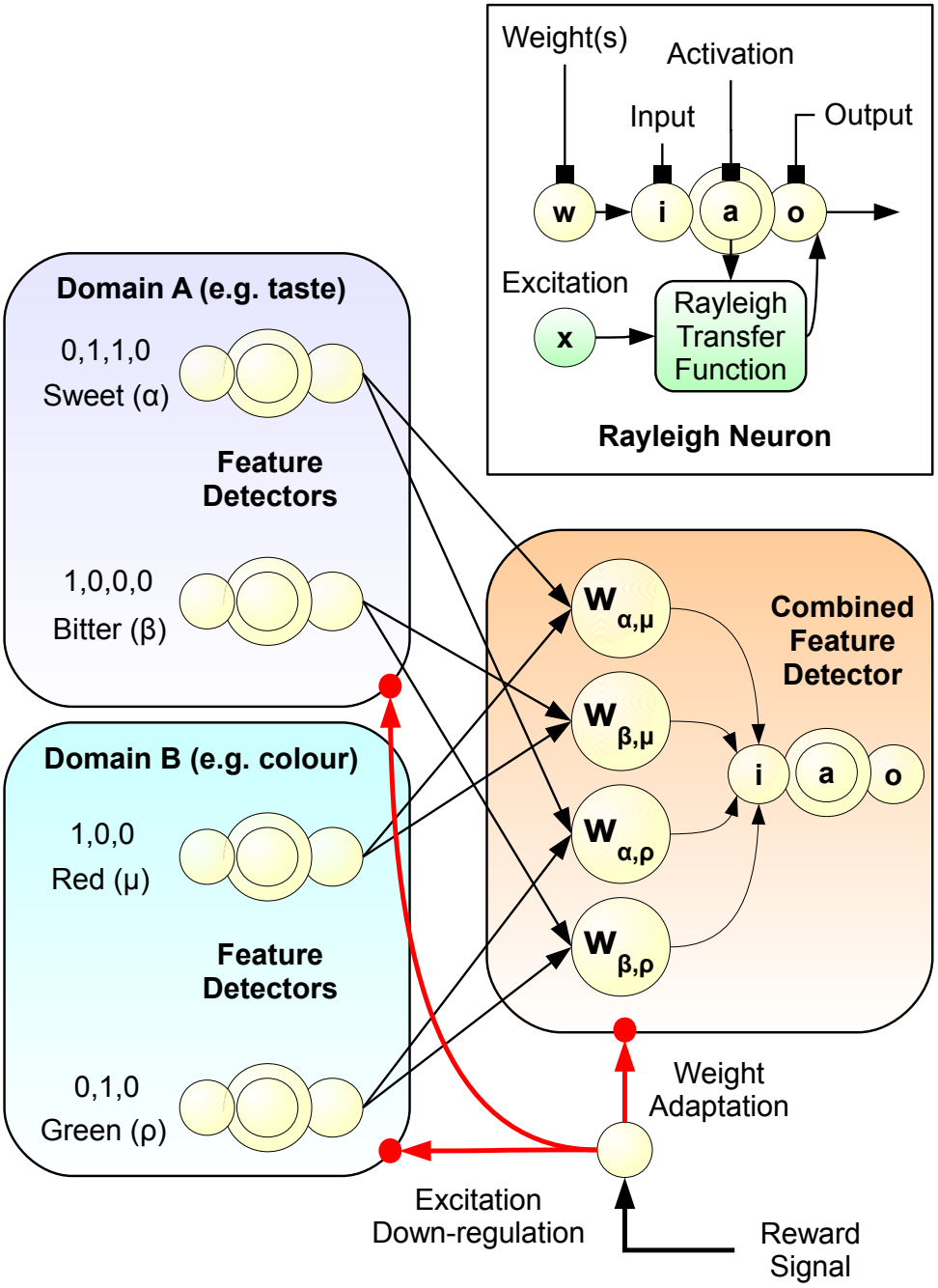


Figure 8.12: **Simple experiment demonstrating energy savings:** Here we assume that there exist two sets of neurons intended for use in feature detection; Domain A (e.g. taste) and Domain B (e.g. colour). The role of the output neuron is to act as feature detector that is sensitive to the combination of inputs that results in a reward. The reward signal has two effects; it adjusts weights to increase sensitivity to the beneficial inputs, and down-regulates the excitation factor of the input layer neurons over repeated reward signals.

of a reward signal. Given the values chose, this corresponds to approximately 10 rewards in a trial comprising 5,000 simulation iterations.

The output-layer corresponds to a single neuron. In contrast to conventional formulations we associate a synaptic weight with each of the possible *combinations* of outputs from the input layer (see figure 8.12), rather than a single weight per input layer neuron output. The goal of the experiment is to adjust the weights of the output neuron so that it detects the particular feature combinations that correspond to a reward, whilst at the same time reducing the total energy dissipated in the transmission of data from the input-layer to the output neuron. Initially we set the excitation values of the input-layer to 0.5, and the weight values to 0.5.

8.4.2 Solution: A two layer network comprising Rayleigh neurons

In solving this problem we devise two simple mechanisms that; to regulate weights so as to detect the reward feature whilst down-regulating the excitation of neurons. The essential features correspond to that of our prior discussions where we down-regulate the excitation of source neurons whilst increasing the receptivity of the target neurons. In this case the source neurons correspond to the input-layer and the target neuron corresponds to the single output neuron.

8.4.2.1 Excitation down-regulation

Every time that a reward pattern is detected, we reduce the excitation factor of the input layer neurons (representing feature detectors) according to the formula;

$$x^{r+1} = x^r - \alpha(x^r - x_b) \tag{8.9}$$

where x^r corresponds to input layer neuron excitation value at the simulation iteration that a reward r is delivered, x^{r+1} corresponds to the new excitation value, x_b corresponds to some base level below which we do not want the excitation factor to drop (taken in our case to be 0.25), and α which determines the rate at which the excitation values change (taken in our case to be 0.33). Simply put,

every time the reward pattern is detected, we reduce the current excitation factor (α)by a value that corresponds to a third of the difference between the current excitation value and the base level x_b. As this difference tends to zero, so does the change in the excitation factor.

8.4.2.2 Weight adjustment

The weight adjustment that takes place in the inputs to the output neuron when a reward is detected, comprising to simple processes. The first is to increase the weight of those inputs that were high relative to the average input, and to decrease the weights of those inputs that contributed less than the average input. In the former case, we limit the weight to a maximum value of 1.0, and in the latter case we limit the minimum value to zero. When a reward signal is triggered, we first calculate the average input value v across all input combinations, based on current input weights w and outputs z from the input layer. Note that the values for z are calculated using equation 8.6, where z corresponds to the firing probability of the Rayleigh neurons that constitute the input layer. Note that the activation value y is set directly as one or zero depending upon whether or not the feature has been detected and x^t represents the excitation factor used in equation 8.6.

$$v = \sum_{n=1}^{N} \frac{w_n(z_j + z_k))}{N} \tag{8.10}$$

In this case N corresponds simply to the total number of combinations of input from the input-layer (in this example 4), w_n corresponds to the weight for a specific input combination (indexed by n). For a given combination n the values of j and k are chosen so that we have a unique combination of inputs from the two sensor domains A and B. Once we have this average value v we adjust the weights according to;

$$w_n^{r+1} = w_n^r + \beta(1.0 - w_n^r) \text{ where } \ w_n(z_j + z_k) > v, \tag{8.11}$$

and

$$w_n^{r+1} = w_n^r - \beta w_n^r \text{ where } w_n(z_j + z_k) < v. \tag{8.12}$$

The value β determines how quickly the weights are adjusted, in our case we took β to have a value of 0.5. The consequence of this weight update rule is that weights for reward combinations increase to 1.0 whilst other weights decline to zero. Note that this update only takes place when a reward has been signalled.

In addition to this weight update rule, we employ a second mechanism to ensure that the activation state of the output neuron is always 1.0 when the reward combination input is signalled. It is this second mechanism that ensures that the output neuron allows generates a high value when the reward input combination is activated, *irrespective* of the absolute levels of the input combinations. It is this mechanism in particular that compensates for a the decline in output values of the input layer in response to reduced levels of activation. The mechanism itself is quite simple, we just normalize individual input values (from different combinations) such that the cumulative value across all inputs sums to 1.0.

8.4.3 Results and discussion

The results of these simulations are shown in table 8.1. The table illustrates the average output level of the input-layer neurons over 5,000 time periods, shown for 5 independent trials. When we use our strategy to down-regulate the excitation factors of the output layer over the course of multiple rewards, we can see that the average activation level is 0.062. This compares will a figure of 0.105 if we fail to down-regulate excitation levels. Assuming a linear correspondence between average activation levels and energy dissipation, then our approach results in a **reduction of energy expenditure by approximately 40 percent**. Without any down-regulation of excitation values, we maintain excitation factors of the input layer at 0.5, however with regulation we are able to reduce the excitation factor from 0.5 to 0.25 whilst ensuring that the output neuron registers all reward combinations. Whilst this toy experiment is very simple, it shows that potential energy savings from such adaptive processes could well be quite considerable.

Table 8.1: Input-Layer Output: Average Activation Levels

Run Index	1	2	3	4	5	Averages
Num Rewards	8	10	9	11	10	9.6
No Regulation	0.106	0.105	0.105	0.103	0.104	(a) 0.105
With Regulation	0.057	0.057	0.066	0.052	0.077	(b) 0.062

8.5 A Basic Circuit exhibiting Tri-Mode Action Selection

On considering the properties of the neuron model, the idea arose that it might help us to solve the exploration problem identified with the conceptual model developed in chapter 7. Whilst we wish to bias action selection choices to pathways that have been demonstrated leading to a primary reward, if we have yet to encounter a reward, we need to explore action selection options in some stochastic fashion. At this point, we temporarily depart form our dopamine modelling attempts, and see how this might help us solve the more general problem of state space exploration through stochastic selection, whilst allowing for prioritisation of option choices that lead to rewards, once rewards have been discovered.

8.5.1 Selection-Exploration-Exclusion

Initially we imagined a dummy experiment comprising a virtual world with a mobile agent. The world comprises 5 locations. By visiting work, the agent can earn money, which would allow the agent to buy food, which they could take home, and eat. Thus eliciting a primary reward. The agent would have explicit sensors for each location, state sensors for money and food, together with indications if the different locations were open or close, and finally an indicator of whether or not it was day or night. Whilst this virtual world concept might seem overly complex, the basic idea is very simple. We want to imagine how the agent would initially randomly explore this world, before subsequently biasing action selection choices to those which would increase the likelihood of the primary reward being attained in subsequent actions. We then relate this to the Neuron Model outlined above.

The results of this thought experiment are shown in Figures 8.13 and 8.14. The key conceptual result of this, was that three modes of activity naturally emerge from this scenario. An intermediate neuron, is created initially when the primary reward (eating food at home) is first experienced. Despite excitation being gradually reduced after repeated rewards, the intermediate node still supports three distinct selection modes; **selection**, **exploration** and **exclusion**. When the same sensor pattern associated with the primary reward is discovered the corresponding action is strongly selected (**selection**). When the sensor patterns only partially match the those associated with the reward, the output of the intermediate neuron is weak, such that any of the actions could be randomly **explored**. Finally, when there is no match at all to the reward associated sensor pattern, the specific action associated with the reward is **excluded** (as its reduced excitation state ensures that its output is less than that of all the other neurons).

8.5.2 Basic Circuit

The simplest system that we can imagine is one which has exogenous inputs (which will call sensor neurons), outputs (which we will term motor neurons), with a single action selection neuron, that supports a winner-takes-all comparison of the signal outputs of the motor neurons. When a reward is discovered, we create an intermediate neuron which has connections with the sensor neurons that were active when the motor neuron was selected. Since we are not able to prejudge how many sensor inputs need to be active either singly or in concert with other sensor neurons to achieve the reward, it is necessary to create a combination of inputs, whenever a primary reward is attained. The next chapter investigates whether or not this implementation is capable of supporting the three action selection modes which are characteristic of the thought experiment described in section 8.5.1.

The basic circuit is illustrated in figure 8.15. The left of figure 8.15 illustrates a simple circuit prior any reward adaptation. The architecture incorporates (square)'sensor' neurons on the left, followed by (round) 'motor' neurons, which innervate a single (square) action selection neuron. The centre section of figure 8.15 shows the changes that occur after a primary reward has been attained. In-

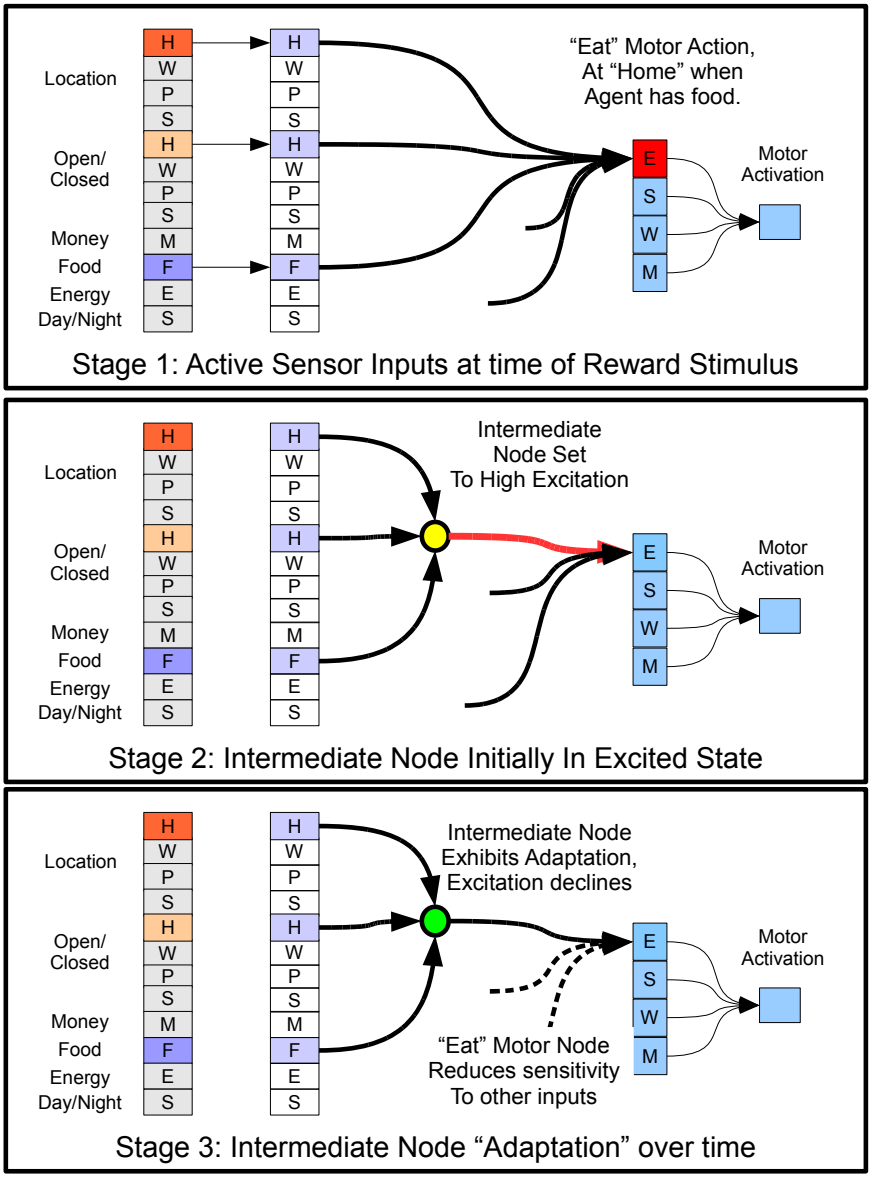


Figure 8.13: **Tri-State Mode Initial Adaptation:**(Top) A primary reward is detected when the agent arrives home with food. (Middle) In response an intermediate is created, initially with the default (high) excitation state. (Bottom) Over subsequent trials, repeated success results in a gradual reduction in the intermediate node's excitation factor. Input Sensor Abbreviations: H, Home; W, Work; P, Park; M, Money; F, Food; E, Energy; S, Day or Night. Output Abbreviations: E, Eat; S, Sleep; W, Work; M, Move.

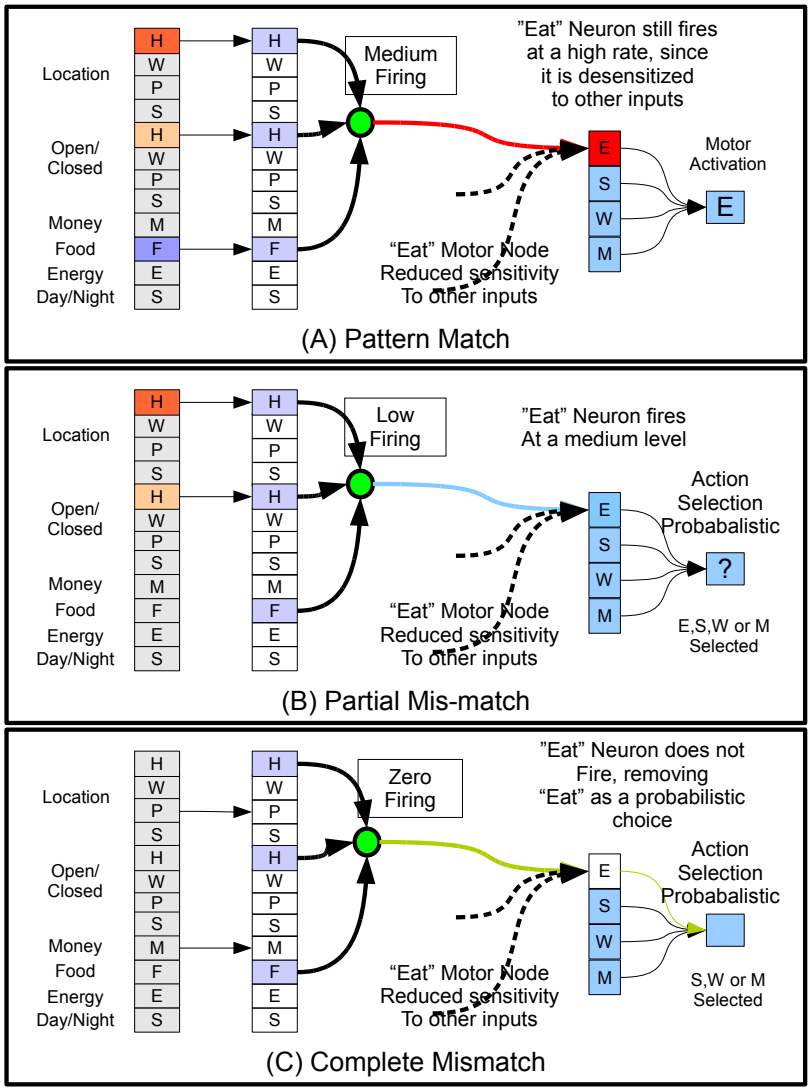


Figure 8.14: **Tri-State Mode Circuit at Run-Time:** At run-time we see three firing patterns, corresponding to three modes **selection, exploration and exclusion**. (Top) If there is a strong sensor pattern match with that of the primary reward, the intermediate neuron ensures that the 'eat' action is **selected**, despite the fact that its excitation state is low. (Middle) With a partial mismatch, the intermediate neuron cannot compete with the other inputs, and stochastic **exploration** ensues. (Bottom) A reward pattern mismatch, will result in a very low firing rate for the intermediate neuron, ensuring that the 'eat' action is **excluded** from consideration. Stochastic selection takes place amongst the remaining options. Input Sensor Abbreviations: H, Home; W, Work; P, Park; M, Money; F, Food; E, Energy; S, Day or Night. Output Abbreviations: E, Eat; S, Sleep; W, Work; M, Move.)

termediate neurons are created, one for each combination of high sensor inputs that are associated with the successful motor action. The right of 8.15 shows how the circuit appears after a number of rewards have been detected under differing action selection sequences. The number of input combinations increases, creating multiple intermediate neurons. This will occur when there are alternative routes to attaining the reward, or where the reward is neutral to certain arbitrary actions.

The process of intermediate node formation is illustrated in more detail in figure 8.16. We create intermediate neurons dynamically every time a primary reward is detected. The sensor neurons that were 'high', when the consequences of the selected action (determined by the winning motor neuron) results in a reward, are linked to the winning motor neuron by intermediate neurons. The intermediate nodes will accumulate as sensor signals that have no predictive power are mistakenly associated with the reward. In our implementation, care is taken to ensure that multiple combinations are not duplicated. Note that in the middle diagram, we create connections which relate to an alternative motor choice. This could only happen if two motor actions resulted in a primary reward signal (i.e. the actions were equivalent in their outcome).

Note that the intermediate nodes as they are presented hear are generated algorithmically. This is not intended to be representative of the dynamic creation of intermediate neurons or interneurons (although we might use that term) in biological systems. Local neurons are interconnected via disperse dendritic trees, such that any single arbitrary *combination* of source neurons have at some level, a corresponding communication channel that in some sense represents a combined signal to a target neuron. Representing such associations between a large number of neurons in software would be prohibitive in both memory and computational costs. Rather we limit ourselves to the simulation of combinations that are implicitly related to the attainment of a reward. Whilst we generated these 'intermediate' nodes algorithmically, the biological analogue is of some strengthening of a combinatorial signal pathway that already has some physical manifestation.

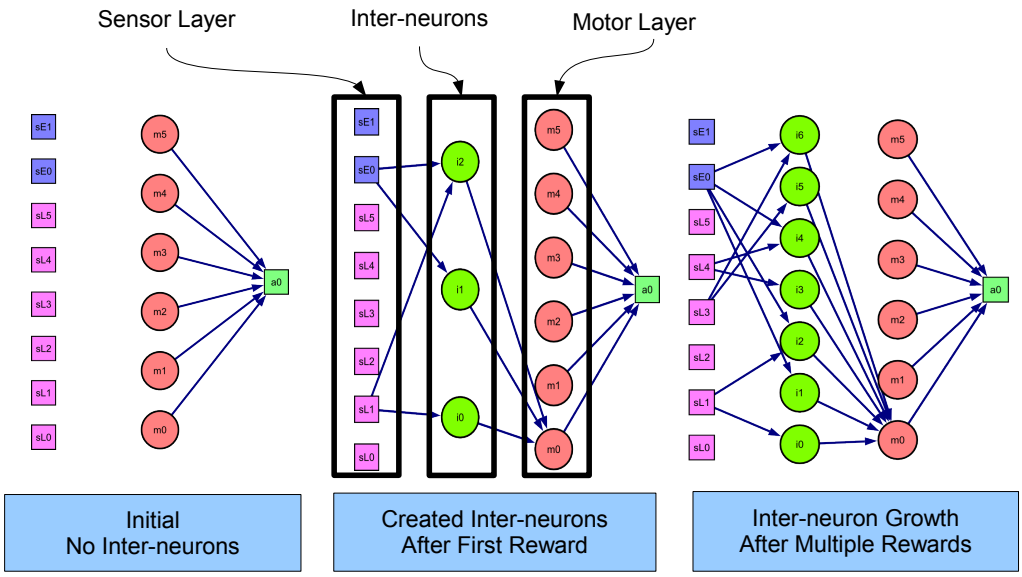


Figure 8.15: **Circuit Basic Idea:** (Left) Our simple circuit starts with (square)'sensor' neurons on the left, followed by (round) 'motor' neurons, which innervate a single (square) action selection neuron. (Middle) When a primary reward is attained, intermediate neurons are created, one for each combination of high sensor inputs that are associated with the successful motor action. (Right) After a number of simulations the number of possible input combinations increase, creating multiple intermediate neurons. This may occur as a sensor input that was high when a primary reward was experience may not be invariant with respect to primary reward elicitation (i.e. it is falsely associated with a selected action).

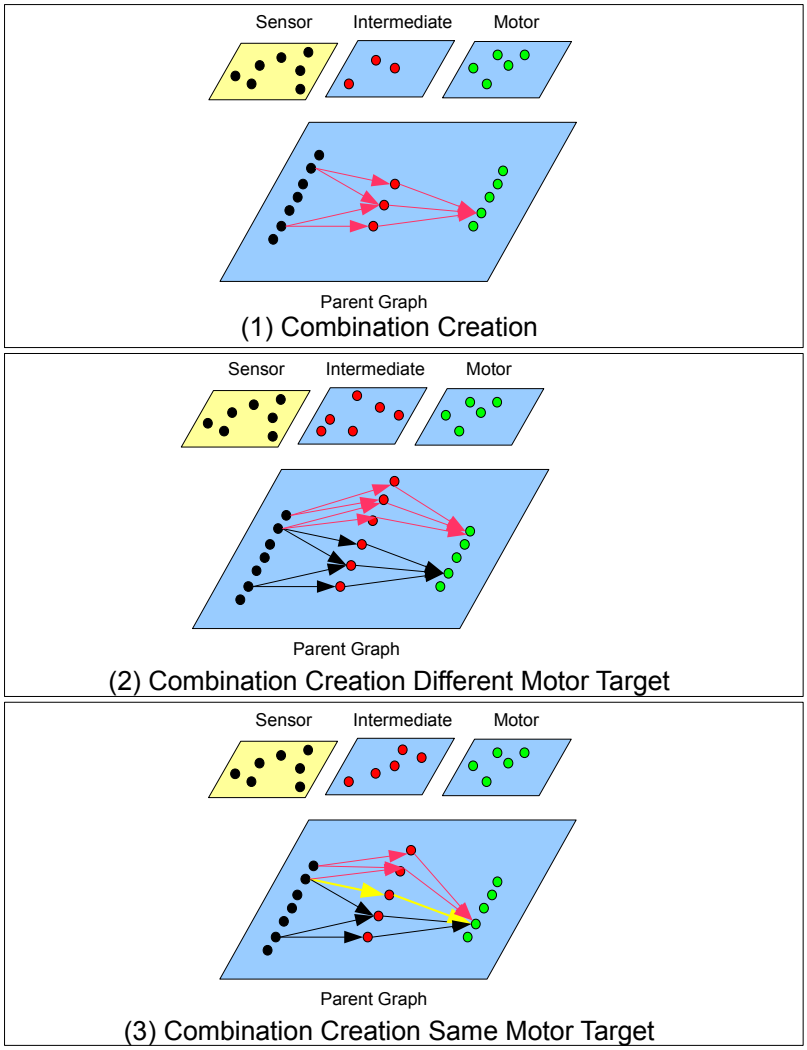


Figure 8.16: **Circuit Intermediate Node Formation:** We create intermediate neurons dynamically every time a primary reward is detected. The sensor neurons that were 'high', when the consequences of the selected action (determined by the winning motor neuron) results in a reward, are linked to the winning motor neuron by intermediate neurons. The intermediate nodes will accumulate as sensor signals that have no predictive power are mistakenly associated with the Reward. In our implementation, care is taken to ensure that multiple combinations are not duplicated. Note that in the middle diagram, we create connections which relate to an alternative motor choice. This could only happen if two motor actions resulted in a primary reward signal (i.e. the actions were equivalent).

Chapter 9

Simulation Environment for Reward Adaptive Circuits

9.1 Introduction

This section describes the common experimental set-up used for the purposes of exploring the incremental adaptation of simple circuits based on the proposed neuron model. Before doing this, some details of the Agent and Neural Graph Library for simulations are presented. This library forms the basis not only of the neural network models, but is also used to support the agent simulations.

9.2 Agent and Neural Graph Library for Simulations

The Lemon Graph library is a mature C++ library of classes written at the Etvs Lornd University, Budapest and at the Budapest University of Technology and Economics. The Lemon Graph library has at its core; *nodes* and *arcs*, with the ability to assign arbitrary data values to each. It supports many common graph algorithms such as; connectivity, shortest path and max-flow algorithms. Although we do not use this functionality, it is conceivable that it could be of some value in the future. We only make use of the graph data structure and the ability to programmatically create nodes and define links between them.

For our purposes the library underwent major augmentation to support *sets*, *groups*, and simulation classes. Nodes that share common features or properties can be aggregated as a group. A new graph class allows for the (hierarchic) insertion of groups into set's. A top-level set represents a domain. A node may have simultaneous representations in multiple groups, sets or domains. These abstractions provide us with a rich vocabulary for building agent based simulations.

The simulation class incorporates one or more *geographies*, and one or more *populations*. Each population member has a *brain*. A geography is a set of location nodes, where arcs represent routes or paths in a spatial domain. A population is a set of agent nodes, where arcs represent relationships (e.g. genetic, social or economic) between agents. A brain is a set of neuron nodes, where arcs represent neuronal connections. The current implementation has been developed to investigate reinforcement learning algorithms in single agents. The functionality of the Lemon graph library is available throughout, with support for the dynamic deletion and creation of nodes.

The simulation class also supports the concept of *assets* or *endowments*. Agents (population nodes) visiting a location (a geographic node) are able to acquire an asset or endowment. This asset or endowment is carried by the agent to subsequent locations. The brain of each agent has sensors (neural nodes) for these assets (and locations) such that neuron state elicits changes in motor state. It is the motor state at the end of simulation period which determines subsequent changes in agent location. The 'brain' domain allows us to implement adaptive algorithms for agent behaviour modification.

The simulation class represents one implementation of a set of algorithms that operate on members of sets and groups over graphs. The separation of simulation functionality from the underlying classes that implement graph, group, set and endowment concepts supports the highly flexible composition of agent simulation implementations.

9.3 Simulation set-up

This section describes the simulation environment used to test the features of model circuits. It has as its features;

1. A graph of locations, which constitute a geography.
2. An agent that is able to visit locations in the geography; either moving to another location or staying in the same place at each simulation time step.
3. Items are situated at some locations which correspond to assets, one of which is reward asset. By visiting an asset location, the agent acquires the asset.
4. An *asset exchange policy* exists, such that if an agent visits an asset location whilst already owning an asset of another type, the asset type that is associated with the location will replace the current asset possessed by the agent.
5. One location is specified as the home location.
6. On returning to the home location whilst in possession of a reward asset, the agent will receive a reward. All assets possessed by the agent on returning to the home location, will then be deemed to be consumed.
7. Each agent possesses a brain, which comprises a graph of neurons, which is used by the agent to determine which location to visit next.

9.3.1 Geography

The computer application that supports this simulation has parameters that control the total number of Geographic locations, and the total number of asset locations. There is only ever one home location, and there is only ever one reward asset location. As we increase the number of Geographic locations, the average reward received by an agent making random location selections will decline. Similarly, as the number of asset locations increases, the average expected reward rate also declines. The greater the number of asset locations, the higher the probability an agent visiting a random location would lose any reward asset that may be in their possession (due to the asset exchange policy).

Figure 9.1 shows two geographies. The top graph corresponds to a world with a total of 6 locations. This comprises 1 home location, 2 asset locations (one of

which is a reward asset location), and 3 Empty locations. The lower graph depicts a geography comprising a total of 10 locations, comprising; 1 home location, 1 reward asset location, 3 other asset locations, and 5 empty locations.

9.3.2 Agents and Assets

As mentioned earlier, there is one asset location where a reward asset is located. By visiting this location, and returning to the home location, the agent receives a reward. Note that if an agent possesses the reward asset, but then visits a different asset location, it will lose the reward asset, as the asset exchange policy ensures that the new (worthless) asset replaces the existing reward asset. However, the asset locations are deemed to be inexhaustible, such that whenever an agent returns to an asset location, there will always be a new asset (irrespective of the loss of a prior asset from this location). Figure 9.2 shows examples of successful and unsuccessful paths through the world. In the top graph the solid arrows show direct paths by which the agent may receive the reward. From any location the agent can visit the *reward asset location*, and then in its next step go to the *home location*. There are also indirect routes by which the agent may receive and reward (examples of which are shown with dotted lines in the top graph). If an agent visits the reward asset location, and then visits any other location *avoiding* any other asset locations before returning to the home location, then the agent receives a reward. Such paths we will term indirect paths. In the bottom graph, we are illustrating the case where even after visiting the reward asset location, by stopping at another asset location, the agent loses the reward asset. Returning to the home location will not result in the agent receiving a reward.

9.3.3 Brain

The brain circuit is based on that outlined in the previous chapter. We have a graph comprising a number of nodes, each of which corresponds to a neuron. See figure 9.3 for a schematic diagram of the brain. There are three major Neuron types; input(sensor), output (motor), intermediate, and action selection. The sensor neurons have two sub-types. The first sub-type is the location sensor. There is one sensor for each location in the world which is 'high' when the agent

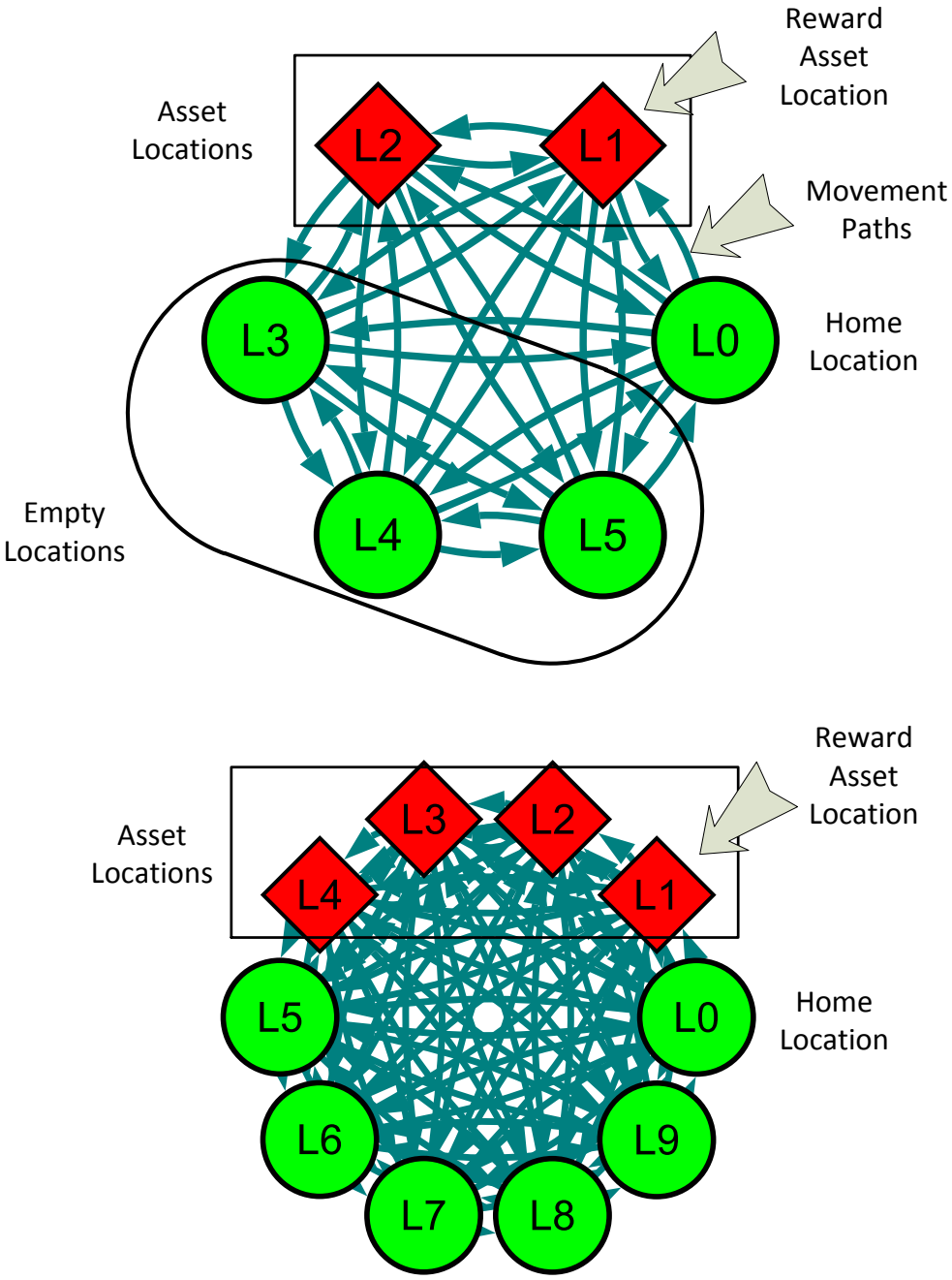


Figure 9.1: **Simulation World geographies: (Top)** A geography with a total of 6 locations. This comprises 1 home location, 2 asset locations (one of which is a reward asset location), and 3 empty locations. **(Bottom)** A geography comprising a total of 10 locations, comprising; 1 home location, 1 reward asset location, 3 other asset locations, and 5 empty locations.

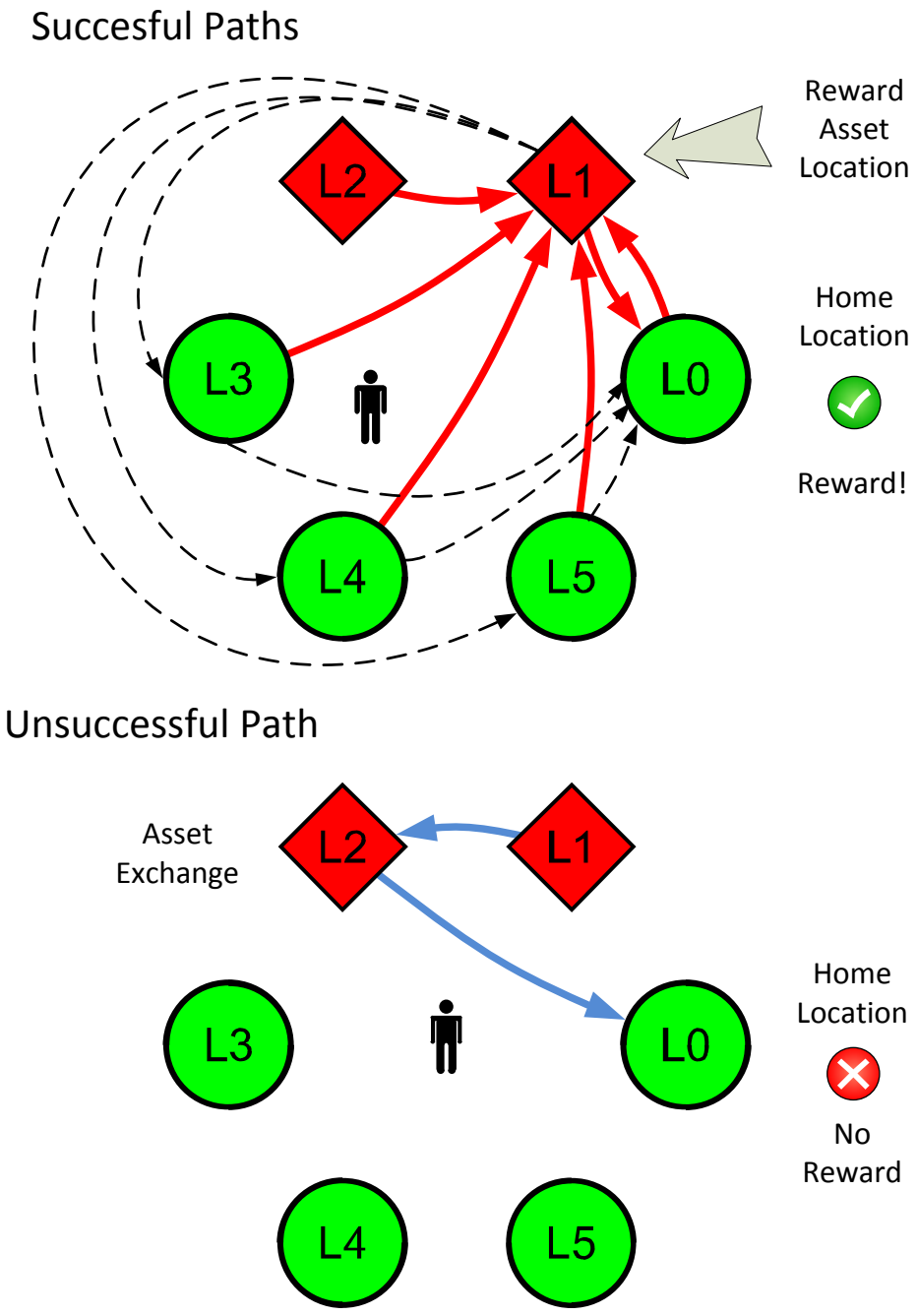


Figure 9.2: **Successful and Unsuccessful Reward Paths: (Top)** The solid lines show direct paths that lead to a reward, the dotted lines show indirect paths, by which the agent is also able to acquire the reward. **(Bottom)** If the agent visits another asset location, whilst possessing a reward asset (acquired at L1), they lose the reward asset. Returning to the home location in this case will not elicit a reward.

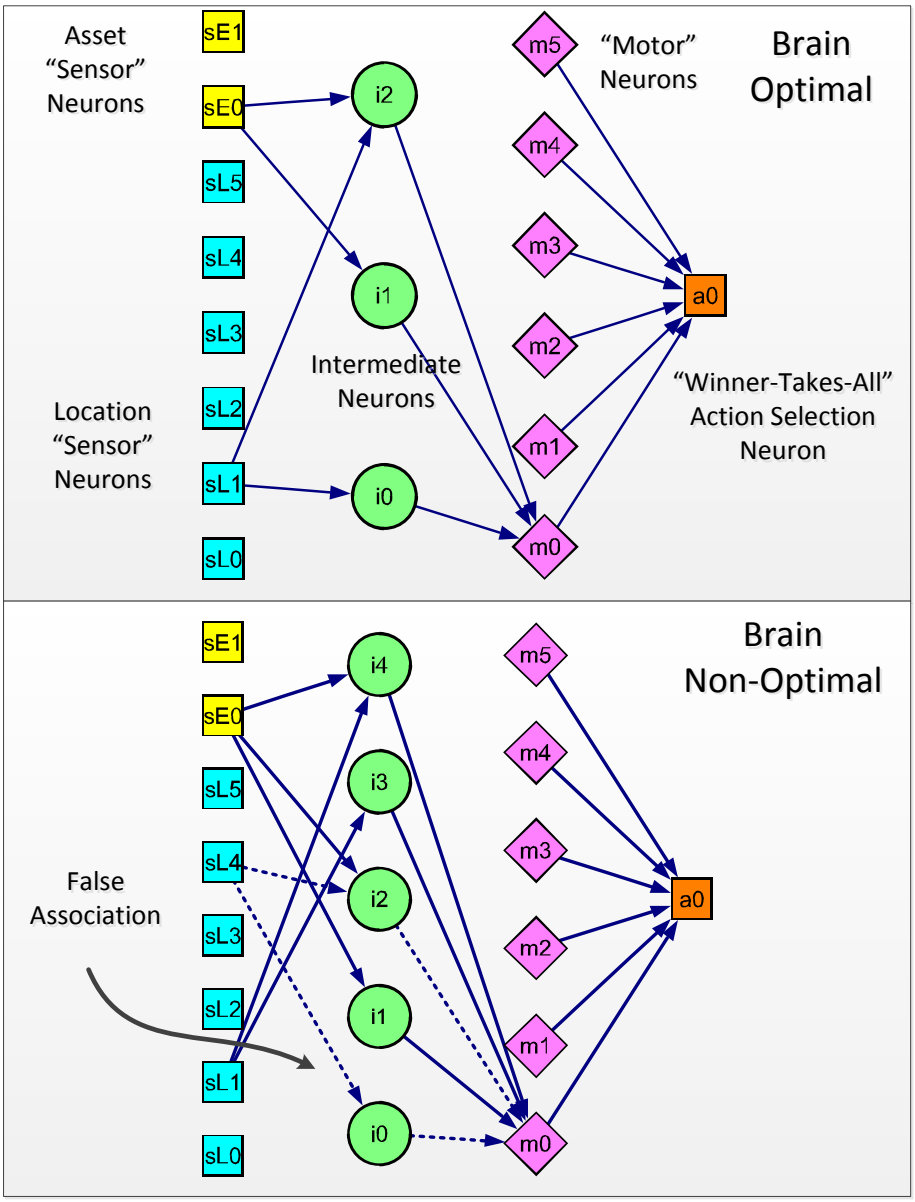


Figure 9.3: **Basic brain Schematic: (Top)** Here we can see an optimal brain, where; **sL1** corresponds to the reward asset location sensor neuron, **sE0** corresponds to the reward asset type, and **m0** corresponds to the motor neuron that selects the home location as the next destination. Subsequently, if the agent is at location L0, or it possesses the asset E0, the **m0** motor neuron will be strongly activated, ensuring that at the next step, the agent returns to the home location, and receives a reward.**(Bottom)** With the existing circuit model, it is possible for false associations to be made. If the **sL4** location sensor is active, this will tend to activate the **m0** motor neuron, biasing the agent to returning to the home location. This will not result in a reward.

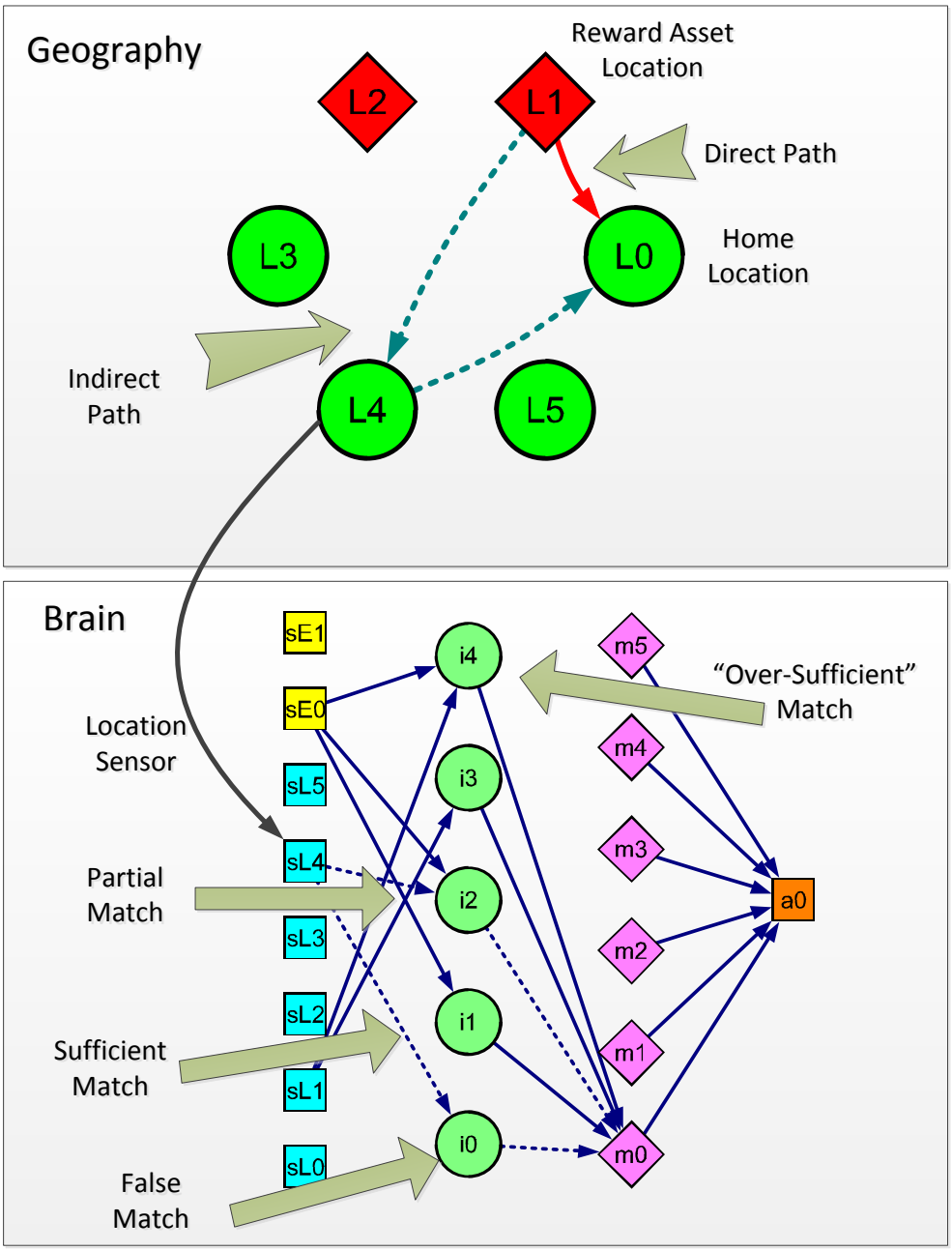


Figure 9.4: **Indirect Reward Paths Create False Associations:** (Top) Here the agent does not take the direct path (solid line), but the indirect path (dotted line), before arriving at the home location whilst in possession of the reward asset. (Bottom) The intermediate neurons **i0**, and **i2** are created as a consequence. The location sensor **sL4** has an input to both these neurons. This means that a bias will be created to activate **m0**, after visiting location **L4**, irrespective of whether or not the agent possesses the reward asset.

is at a location that corresponds to the sensor. The second sub-type is the asset sensor. There is one asset sensor for each asset type in the World. There is no specific sensor for the reward asset, as the type of the reward asset is unknown to the agent. The motor neurons compete for the decision as to which location should be visited next. There is, as a consequence, one motor neuron per world location. The role of the Action Selection neuron is simply to look at each of the motor neurons, and determine which has the highest output. The motor neuron with the highest output at each simulation step, determines which location that will be visited next. The Model Neuron described in the prior chapter is used for the motor neuron.

Each time the agent receives a reward, new neurons are created. These are the intermediate neurons (or nodes). When a reward is detected (by the agent returning home, whilst in possession of the reward asset), those sensor neurons that were high, when the successful motor neuron was selected, are connected to the motor neuron by Intermediate nodes. The Intermediate nodes created, correspond to one for each sensor neuron *combination*, comprising a power-set of all connection combinations between the high sensor neurons and the selected motor neuron.

Note one important aspect of the model. When a reward is received by the agent, the excitation factor of the winning motor neuron can be adjusted. Down-regulation will on subsequent simulation runs, have the effect of inhibiting the output of this motor neuron, compared to other competing neurons (thus resulting in the **exclusion** of this action choice). Unless the inputs to this neuron are sufficiently activated, it will lose in competition against motor neurons with no input Intermediate nodes. If the input signals to the motor neuron are high enough, then the output of this motor neuron will be higher than the other motor neurons, ensuring that the corresponding action is taken. This corresponds to the **selection** mode in our tri-state mode selection model described in the previous chapter. Note that in the initial stages, prior to any reward being received, all motor neurons compete stochastically, resulting in *exploration* of the action selection options. This results in the world being explored. Even after a reward has been received, (unless the inputs to the prior winning motor neuron are substantially high), *exploration* takes place between the non-winning motor

neurons, as their excitation factors remain high. Thus, even after a reward has been received, **exploration** will still take place.

Figure 9.4 shows how false associations can be made. An agent could arrive at the home location, possessing the reward asset by an indirect path. In doing so, a set of Intermediate nodes will be created whose inputs comprise a combination of the **sEO** (asset sensor for the reward asset) and **sL4** (location sensor for location **L4**). Thus, a false association will be made between location **L4**, and the motor neuron **m0** that selects the home location **L0**, as the next action.

Finally it is worth mentioning that we do not *explicitly* model the signally that takes place in dopaminergic neurons. The adaptation that takes place in the dynamic creation of neuron connections and the adjustment of neuronal excitation values are deemed to be *implicit* effects of dopamine modulation.

9.3.4 Simulation Application

Figure 9.5 shows a schematic of the simulation loop. Figure 9.6 shows a screen shot of the simulation application. Key features of the simulation application include certain real-time visualization features;

- Neuron activation state, and input weights.
- Dynamic creation of new intermediate neurons and connections.
- Automatic layout of new circuit configurations.
- graphs of dopamine tonic and phasic activity.
- Numeric display of neuron inputs and Outputs.
- Neuron oriented: connections coloured by activation state, internal neuron state coloured.
- Connection oriented: neuron weights and activation states coloured in the connections view.

In addition to this we have the facility to;

- set simulation breaks, so that we can inspect the network on reward triggers.
- carry out batch simulations automatically.
- allow the user to adjust all simulation parameters from the parameter grid.
- generate all simulation and batch metrics and have them displayed in the results grid.

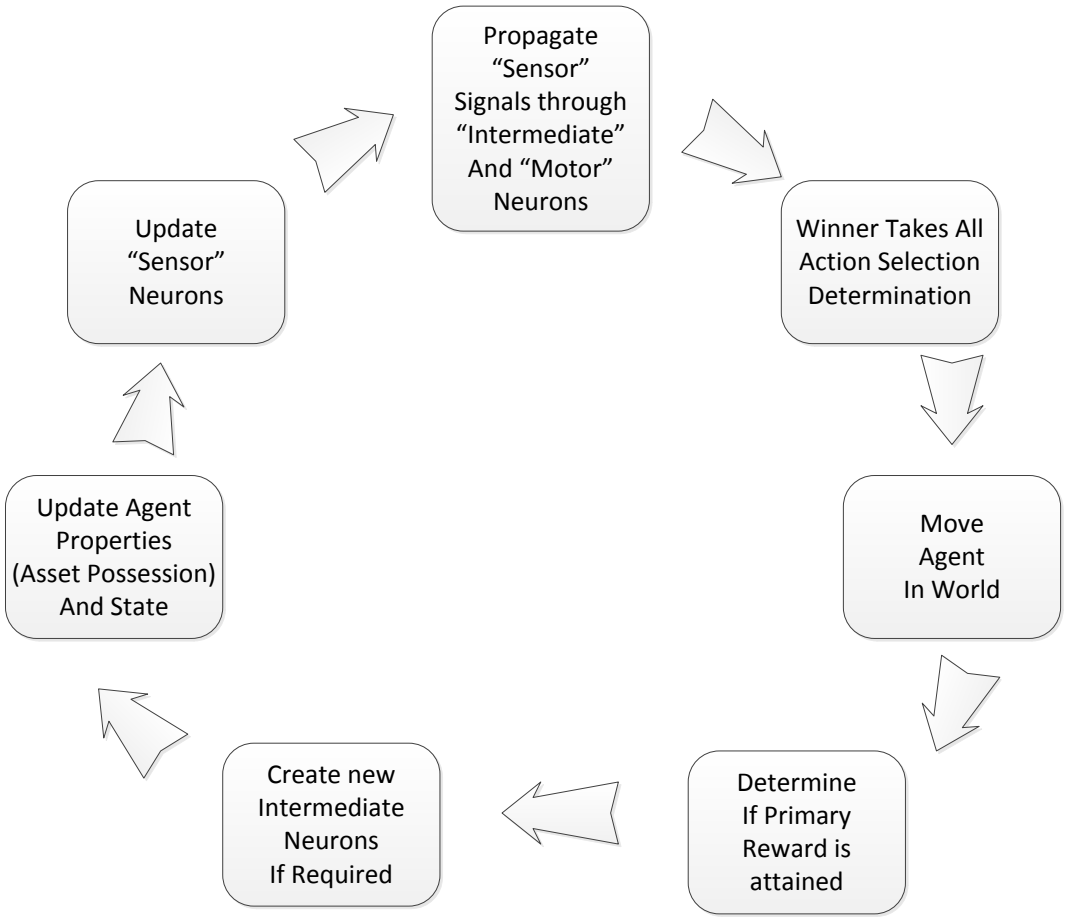
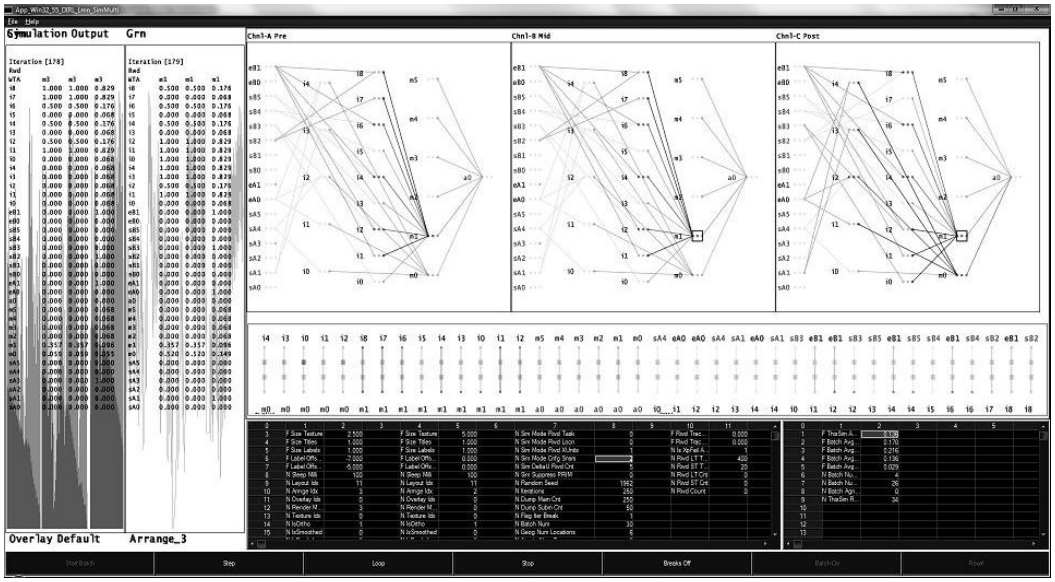


Figure 9.5: **Simulation Schematic**



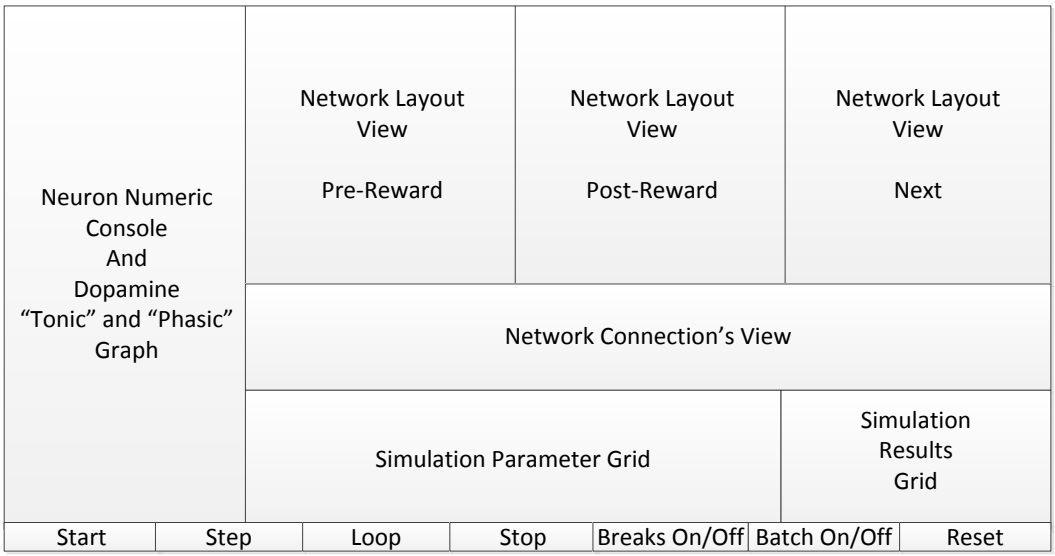


Figure 9.6: **Simulation Application Screen Shot:** (Top) Here we have a screen-shot of the simulation Application. Network state is displayed in real time in OpenGL windows (the screen-shot is shown in inverted grey-scale in print, in colour in pdf format).(Bottom) Here we have a schematic that shows the function of each window in the screen-shot.

9.3.5 Incremental Changes

Whilst this chapter describes the essential features of the simulation environment and the characteristics of Intermediate node creation, its description is limited to that of the basic circuit outlined in the previous chapter. The next chapter seeks to make incremental changes to the basic circuit, and through simulation show how these incremental changes result in enhanced fitness of the agent.

Chapter 10

Model architectures and results

10.1 Introduction

This chapter describes a number of circuit architectures based on our new neuron model. We start with a very simple network structure, and through incremental changes develop increasingly complex structures. Through simulations that employ the world model described in the previous chapter, we are able to identify the functional characteristics of these circuits. The direction of the incremental changes in circuit complexity represent a manual exploration of circuit variants that could be parametrized for the purposes of supporting evolutionary search; the number of layers in each circuit, and the particular algorithm used for adapting the network on receiving a reward. The 'fitness' of each circuit variant relates to essentially the same task, that of exploring a number of action selection options, and based on the presentation of a reward signal, exhibiting adaptation that biases future actions in favour of repeating those that have historically been associated with the discovery of the reward. This allows us to compare the relative strengths of each circuit using a common base.

The simulation environment consists of 6 locations, comprising; 2 asset locations (one of which is the reward asset location), 1 home location and 3 empty locations. The output of the action selection neuron (a common feature of all the circuits), corresponds to the result of a 'winner-takes-all' algorithm, determining subsequent agent movement. More complete details of the simulation task are

described in the chapter 9. Note that all figures in this chapter that depict neural networks (other than the schematic illustrated in figure 10.3 that depicts the architecture of a delta unit) represent results generated as part of a simulation run.

10.2 Incremental Stages

10.2.1 Stage 1: Single Layer

This is the basic circuit described in chapter 6. It incorporates input ('sensor') neurons, output ('motor') neurons, intermediate neurons created as a result of a rewards. The intermediate nodes are created dynamically, and associate sensor states with the winning motor neuron when a primary reward is detected. A single action selection neuron uses a winner-takes-all algorithm to select the winning motor neuron at each time step. Subsequent agent movement is determined by this winning neuron.

By default, we implement a primitive *failed-expectation* algorithm that reduces the weights of the inputs to the winning motor neuron, when its activation failed to result in an anticipated reward. All weights start at 1.0. The weight reduction factor is arbitrarily chosen to be 0.5. On a failed expectation, new motor input weights are calculated by multiplying the input signal with this weight reduction factor for each input to the winning motor neuron. In the summary section, results are shown with and without this failed-expectation weight modification.

10.2.2 Stage 2: Dual Layer

The Dual Layer model corresponds to a duplication of the input stage of the basic circuit model. sensor neurons are duplicated. intermediate neurons are created via the same process as that which governs the creation of intermediate neurons in the basic circuit. Duplicate intermediate neurons are created, one for each combination of the active sensor-motor pairs, when a reward is received. Two examples of Dual Layer circuits are shown in figure 10.1.

At face value, such an exercise may seem to have little value. However, it

is of interest to us as unit duplication is seen as potentially one of the key processes by which evolution in the brain takes place through mutation (Striedter [2005]). Our goal was to see what impact, if any, this has on the fitness of the agent. Specifically we are interested to see whether or not a duplication of input units interferes with the functional operation of the basic circuit. If it does not, then we have a potential evolutionary pathway, by which the basic circuit can undergo complexification, as an intermediate step towards the more complex circuits described later in this chapter.

10.2.3 Stage 3: Temporal Units

If we have a circuit that comprises duplicate sensor units, we can imagine a mutation whereby the output of one sensor unit, becomes the input of the duplicated sensor unit. With a suitable latching signal we could generate a memory buffer capable of storing sequential sensor patterns. This is the idea behind our temporal unit circuit. We simply latch the contents of the current sensor inputs into a secondary sensor unit at the termination of each simulation interval. When a reward signal is detected we see the creation of two sets of intermediate nodes. The first set of intermediate nodes connects the current active sensor units with the current winning motor neuron. The second set of intermediate nodes connects the prior active sensor units with the prior winning motor neuron. Two examples of circuits incorporating these features are shown in figure 10.2.

10.2.4 Stage 4: Delta Units

Two examples of circuits incorporating a delta unit are shown in figure 10.5. Once we have an arrangement of duplicate sensor units, whose contents are temporally delayed we can consider a further adaptations. Let us assume that some state variable exists that acts of a metric of the number of times that an individual sensor neuron is active when its associated motor neuron elicits a reward. Associations are captured by the intermediate nodes that are created in response initially to the detection of a primary reward. By comparing this metric for all sensor inputs associated with a winning motor neuron, we are able to identify which of the inputs appear to contribute most significantly to the activation of

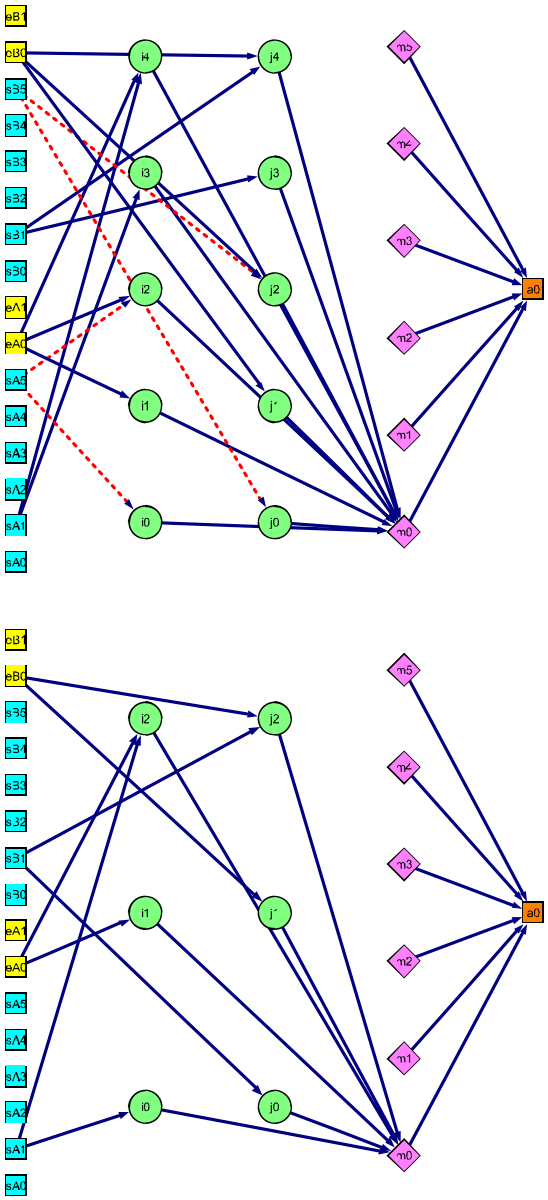


Figure 10.1: **Stage 2: Dual Layer (Top)** A Dual-Layer circuit with false associations (dotted lines). **(Bottom)** A Dual-Layer circuit, corresponding to the optimal single layer circuit. The agent chooses to return to the home location **L0**, when either in possession of a reward asset or being at the **L1** location. The signals provided by the Dual-Layer versus the Single Layer circuit are redundant, however agent fitness is broadly similar.

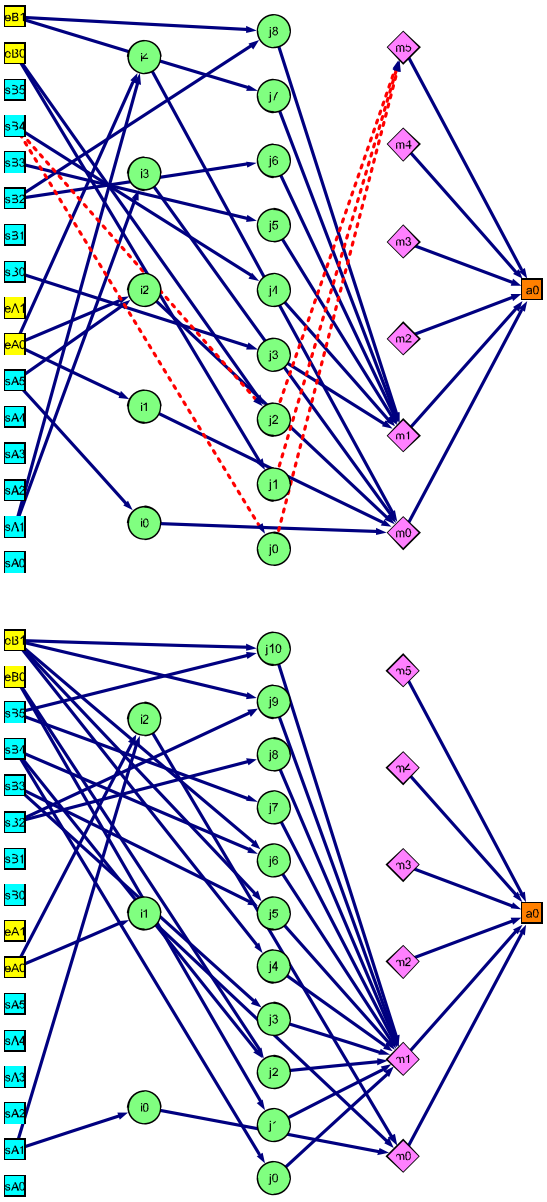


Figure 10.2: **Satge 3: Temporal Units (Top)** Here we have two sensor Units. The lower sensor Unit has inputs that are current. The upper sensor Unit has inputs that are delayed. We can see that the temporal unit has made a false association with the **m5** motor neuron, however this does not irrevocably damage performance. **(Bottom)** A more efficient example of the same circuit is shown here. All the temporal unit outputs lead to the selection of the **m1** motor neuron. This corresponds to a movement to location **L1**, where the reward asset is located. The result of this is that the agent enters a loop, going from the home location (at **L0**) to the reward location **L1**, and back again.

the winning motor neuron that successfully elicits a reward. Let us call these sensor neurons, *significant* sensors.

The next key step, is to generate a signal that is activated when we see a transition in the output of one of these significant sensors from zero to one. This signal is responded to as if it was a reward signal in itself. We term this an "implied reward" or Conditioned Stimulus trigger. This "implied reward" or Conditioned Stimulus trigger can then be used in place of the primary reward signal, to trigger the creation of intermediate nodes.

In the basic circuit, a primary reward signal elicits the generation of intermediate nodes that associate the current sensor state, with the winning motor neuron. In the temporal form, a primary reward signal elicits the same response. In addition, associations are also created which link the prior sensor states, with the prior winning motor neuron, to create in effect associations which support a sequence of motor activations.

The delta unit form of the circuit responds in a similar fashion to that of the basic circuit in its response to the first instance of a primary reward. However, we do not use additional sensor buffer stores for storing sequential data. What we do is to detect when we have a positive state in one of the significant sensor neurons. We then use this as an implicit reward signal to trigger the creation of intermediate nodes that associate the winning motor neuron that precipitated this transition, with the sensor neurons that were active contemporaneously. The effect of these newly created intermediate nodes will be to bias future motor selection in favour of actions result in the activation of one of the significant sensors. Whose activation in turn, promotes the motor action associated with the primary reward.

The temporal unit form of the circuit is effective at learning motor actions that should be executed in sequence (depending upon prevailing sensor inputs) for the attainment of a prior reward. In contrast, the delta unit form responds to significant sensor inputs, irrespective of whether or not the sensor was activated immediately prior to eliciting the primary reward, or multiple time periods before this. It does not respond to the strict time or sequence order in which sensor signals are activated. Rather it responds to the average contribution to a primary reward that sensor neuron elicits, and seeks to bias actions towards repeating the

activation of this significant sensor neuron. This circuit form is ideally suited to creating sensor-motor associations when prior actions contribute to a reward, but do not necessarily have to take place in a strict temporal sequence..

A schematic for this circuit is shown in figure 10.3.

This circuit modification is motivated by the phenomena observed in dopamine signalling where the peak of activity migrates from the primary reward to that of the prior Conditioned Stimulus. The results of this effect can be seen in figure 10.4. In the terminology of our model, we imagine that after stabilization of weights associated with the primary reward, a mechanism exists, whereby changes to the inputs of the sensor neurons associated with the selection of the primary reward, themselves trigger a Delta Signal. The Delta Signal itself triggers the creation of intermediate neurons *as if* a primary reward signal had been generated. In this case however, we are creating associations between the motor neuron that elicited changes to the primary sensor Unit key neurons, and the sensor inputs that were active when this motor unit was triggered.

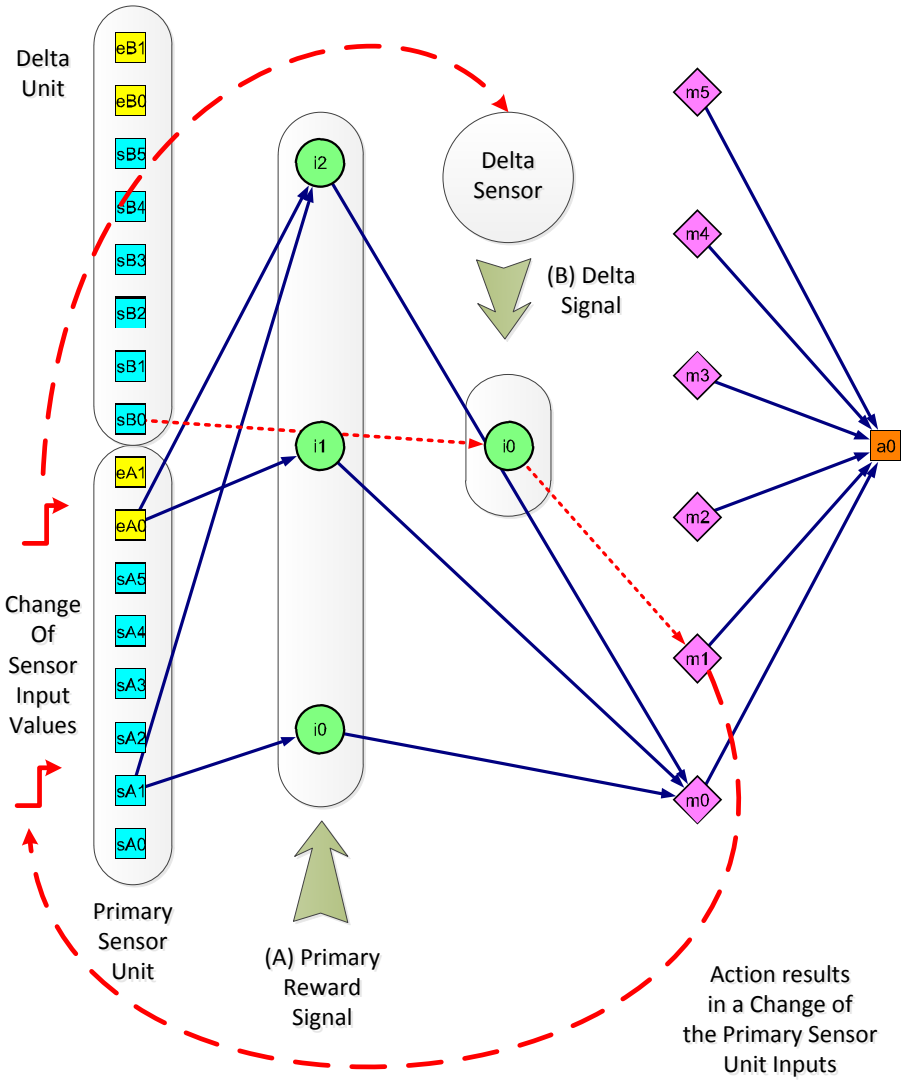


Figure 10.3: **Delta Unit Schematic:** Initially a **primary reward** signal associates the winning motor neuron, with combinations of the sensor neurons that were active when the reward was triggered. This is done through the creation of the intermediate nodes connected to the primary sensor Unit. After a period of adaptation, the weights of the inputs to the motor neuron **m0** will stabilize. The inputs that make the greatest contribution to the selection of this motor neuron, are monitored by a delta sensor. The delta sensor generates the **delta signal** that stimulates the creation of intermediate neurons, in a fashion analogous to that of the **primary reward Signal**.

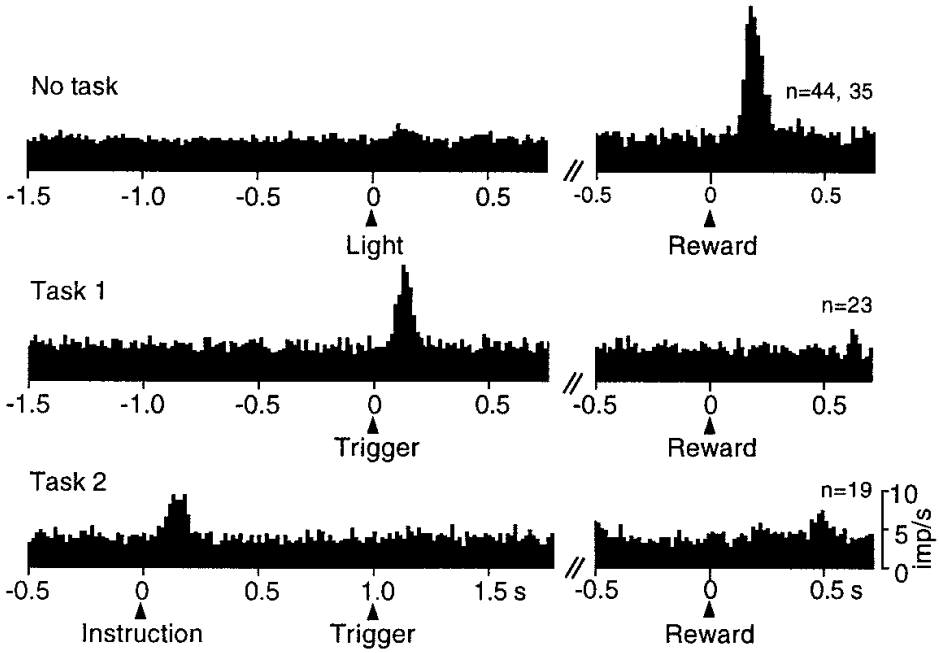


Figure 10.4: **dopamine phasic activation shifts to the earliest event in a reward chain**The charts correspond to time binned histograms of dopamine phasic activity in three seperate experiments. Both the instruction and trigger correspond to conditioned stimuli. What is important to note, but what is not shown here, is that for a transient period high activation levels are coincident with **both** the reward presentation and the presentation of the conditioned stimulus. Source Schultz [1998].

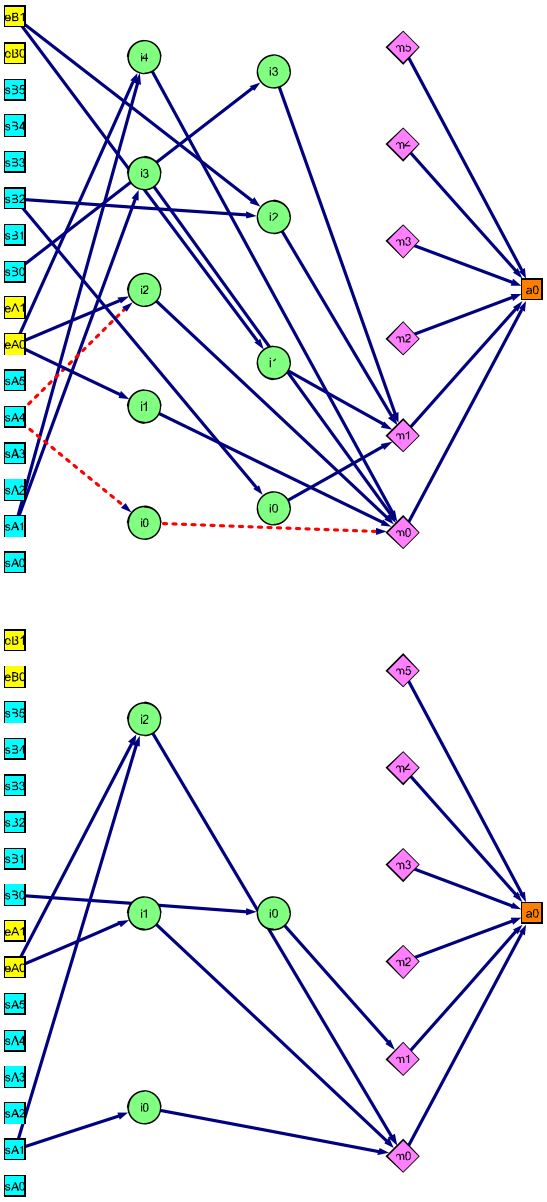


Figure 10.5: **Stage 4: Delta Units (Top)** A reward circuit with a primary sensor Unit (lower) and a Delta sensor Unit (Upper) A false association is shown with a dotted line. **(Bottom)** This corresponds to an optimal solution, but contrasts significantly with that of the temporal unit circuit, in that very few connections are made. The delta unit has identified a single sufficient sensor condition to ensure that the agent Loops between the home location at **L0** and the reward location at **L1**.

10.3 Results Summary

The results are summarized in Tables 10.1 and 10.2, together with the schematic shown in figure 10.6. These represent the result of 30 trials for each circuit type, with each trial comprising 250 simulation iterations. The schematic shown in figure 10.6 is indicative of potential dimensions of evolutionary exploration, such as; the number of 'layers' in a circuit, and the reward adaptation algorithm used (e.g. for strict temporal sequence learning, or delayed rewards).

The maximum theoretical average reward score is 0.5. This corresponds to an action sequence forming a loop between the home location and the reward location, which results in 1 reward for every two simulation steps. The average reward due to the random selection of actions at each iteration is 0.06. We use this value as our base value for converting average reward values to relative fitness values; an average reward value of 0.06 corresponds to a relative fitness value of 1.0.

The delta unit circuit form yields the highest average reward value of 0.38, a relative fitness value of 6.3 (i.e. the delta unit form yields rewards 6.3 times greater than that which would be achieved due to random action selection). This is shown as entry (f) in table 10.1, where the excitation factor of the motor neuron remains unchanged at a value of 0.5 even after a reward has been attained. The delta unit average reward corresponds to 76 percent of the maximum theoretical value. In considering the complexity of the circuits generated (see figures 10.1, 10.2 and 10.5), we can see that the delta unit generates the simplest networks.

Table 10.2 compares results where we adjust the weights of an input to the motor neuron due to a failed expectation. This was initially thought to be a useful feature of the model circuit, but the results indicate that it does not have a substantial effect in this set of experiments. Whilst it is generally beneficial, it marginally reduces the total fitness of the delta unit that has an excitation of 0.47. By inspection the reason appears to be that false associations result in the agent returning to the home location, even when not in possession of the reward asset. Whilst this may appear to have a negative impact, this increases the probability of the agent arriving at the home location. From there, prior associations ensure that the agent returns to the reward location. Under these

circumstances, returning early to the home location does not have a negative impact on the average rewards attained.

Changes to the excitation factor of the winning motor neurons (when implicated in a reward sequence) did not the anticipated effect. We hypothesized that when a motor neuron is implicated in a reward sequence, a reduction in its excitation factor would help to ensure that it would be excluded from future action selections unless its inputs were strongly activated. It was imagined that this would support a more efficient exploration of available action choices under circumstances where exploration of the choice of actions was in some sense incomplete. Whilst the logic of this hypothesis remains valid, the results for the temporal unit form of the circuit show that this can reduce the average reward payout. The optimal solution is one which forms the smallest loop between the home location and the path that visits all the reward locations necessary to trigger the primary reward. Whilst the temporal unit is able to discover the action sequence that leads to the reward, there is no automatic closure of the loop. Closure of the loop requires that the action selected from the home location ensures that the agent returns to the start of action sequence that results in triggering the primary reward. Whilst it is possible for the temporal unit to discover this loop closure (assuming that there are sufficient layers in the temporal buffer), this typically does not take place at the very first attainment of the primary reward, since it is achieved through random discovery. It is only due to subsequent exploration from the home location that this loop closure takes place. Let us now consider two cases where an initial primary reward has been attained (resulting in circuit adaptation that will increase the probability of the same reward action sequence being repeated), but the loop has yet to be close. In the first case denoted by (a) in table 10.1), the excitation factors of the motor neuron nodes implicated in the reward chain are reduced from 0.5 to 0.45. Initially from the home location, no strong association exists that will ensure that we return to the start of the reward sequence. Random action selection from the home location could therefore result in the agent visiting a location that is not involved in the reward chain. However, once this occurs, given that the excitation factors of those motor neurons that are implicated in the reward chain are down-regulated, those actions which result in a return to the home location are inhibited. This can result in a reduce

probability of returning to the home location, and discovering the Loop closure. Let us compare this with that of (b) in table 10.1, where the motor neurons excitation remains unchanged at 0.5, even after it is implicated in a reward chain. Under these circumstances, the probability of re-visiting the home location from an arbitrary node that has no pre-existing associations is as high as it is for any other node. This ensures that the motor activity that results in a return home location is not excluded from the exploratory process. This increases the chance of returning to the home location, and from there, discovering the action selection choice which results in a closure of the optimal reward path, creating a loop. The effect of down-regulating motor neuron excitability in this case, works against the discovery of this optimal path. Similarly, a significant reduction in the excitation factor of the motor neuron, when implicated in a reward sequence, also reduces the average reward in the case of the delta unit form or the circuit.

In conclusion, both the delta and temporal circuit formats yield respectable results, 75 percent and 66 percent respectively respectively. 100 percent is not attainable since there is a necessary period of exploration required to discover the optimum reward path. However, the optimal results are achieved without the need for the down-regulation of motor neuron excitation levels.

The primitive Single and Dual-Layer circuits (as a reminder the Dual-Layer form is simple a redundant duplication of the Single-Layer form) attain lower levels of relative fitness. However, such circuits could act as intermediary steps in the emergence of the temporal and delta circuit forms, from an evolutionary perspective since still improve relative fitness despite their simplicity.

Table 10.1: Summary of Circuit Fitness

Circuit	Avg. reward	Rel. Fitness
Random	0.06	1.0
Single-Layer	0.15	2.6
Dual-Layer	0.13	2.3
(a) Temporal (Ex. 0.45)	0.21	3.6
(b) Temporal (Ex. 0.47)	0.33	5.8
(c) Temporal (Ex. 0.50)	0.33	5.8
(d) Delta (Ex. 0.45)	0.18	3.1
(e) Delta (Ex. 0.47)	0.36	6.1
(f) Delta (Ex. 0.50)	0.38	6.3

Table 10.2: Expectation Fail Weight Adjustment

Circuit	Avg. reward Expect. Fail Alg. ON	Avg. reward Expect. Fail Alg. OFF
Random	0.06	-
Single-Layer	0.15	0.14
Dual-Layer	0.13	0.13
Temporal (Ex. 0.50)	0.33	0.33
Delta (Ex. 0.45)	0.18	0.17
Delta (Ex. 0.47)	0.36	0.38

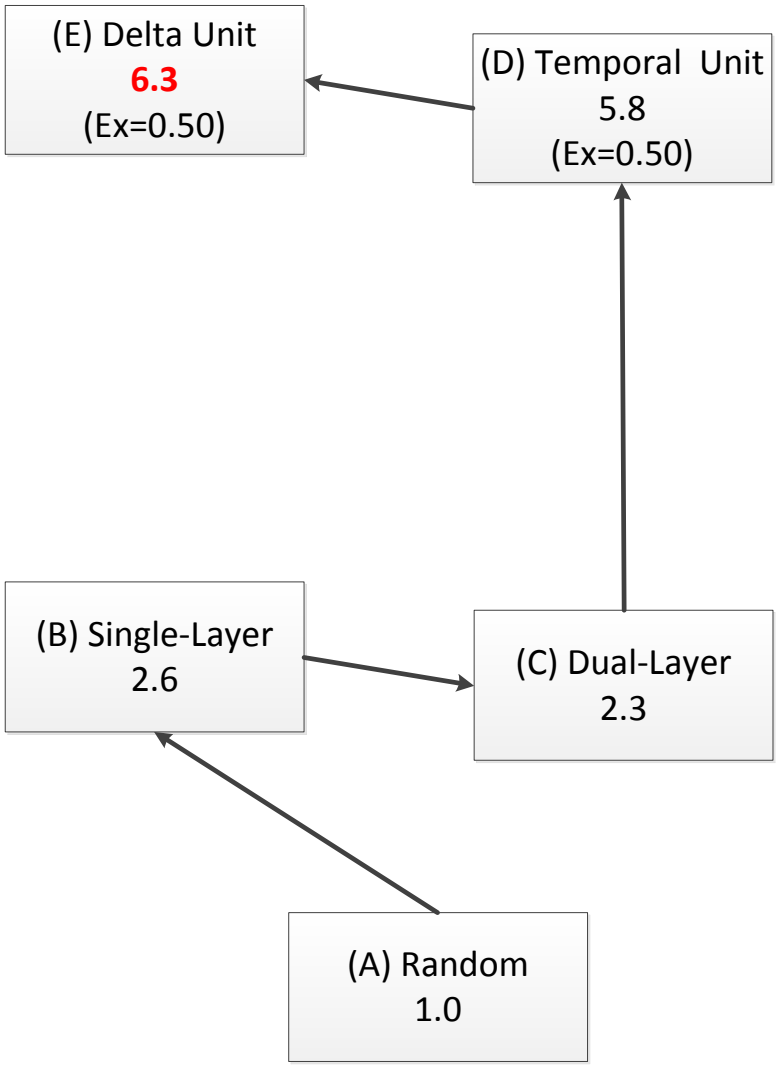


Figure 10.6: **Relative fitness along hypothetical dimensions of evolutionary exploration:** **(A)** agent making random choices has a fitness of 1. Expressed in terms of Relative Fitness we can compare each of the incrementally modified circuits. **(B)** A single-layer circuit could conceivably mutate to one with **(C)** dual-layers, without significant changes in fitness. **(D)** A subsequent mutation which supports the sequential latching of sensor data is able to learn simple sequences. A modification of this circuit, where a Delta Signal denotes a change in the key inputs to the primary reward Unit, supports the backward induction of beneficial sequences, although performance is sensitive to Excitation Factors **(E)** and **(F)**.

10.4 Model Extensions

The simulations described above generate a reward when an agent collects an asset at a single reward location and then returns to the home location. An obvious question to consider is whether or not this model extends to arbitrary sequences. As a preliminary investigation into this issue an extension of the temporal configuration of the circuit was built. Rather than having two temporal buffer units, the circuit contained three. We then carried out a number of batch simulations comprising 30 simulations, each of 500 time steps (compared to 250 time steps used in the original experiments);

- **Redundant temporal unit:** This is a repeat of the experiments described above, but using a temporal form of the circuit with three rather than the original two buffer units. Note that the optimal solution only requires two buffer units.
- **Three Layer Temporal Units with Excitation Regulation:** A primary reward is triggered when locations 2,1 and 0 are visited in strict sequence. motor neuron excitation is down-regulated when a motor neuron is implicated in a reward sequence.
- **Three Layer Temporal Units without Excitation Regulation:** A primary reward is triggered when locations 2,1 and 0 are visited in strict sequence. motor neuron excitation is left unadjusted when a motor neuron is implicated in a reward sequence.

The results of these experiments are described below. The only modification necessary was to introduce a refractory period, whereby a winning motor neuron was inhibited from firing in the subsequent time-step. Under the original model it is possible for a winning neuron to be activated repeatedly due to stochastic effects. When we have a three temporal buffer units and only require the agent to visit two locations to attain a reward (e.g. through visiting a reward location at index 1 and then returning to the home location at index 0), it is possible for self-associations to be created (e.g. of the order of location indices in the agent path is 1-1-0). This is problematic, since a sensor state indicating that the agent

is at location 1 will be associated with a motor action to visit location 1. This can create a loop, whereby the agent is stuck at location 1. To avoid this we set the activation of a winning motor neuron to 0 at the subsequent time step. This ensures that the same motor neuron will not be selected in consecutive simulation time steps, and preventing the creation of this loop.

10.4.1 Redundant temporal unit

The first of these experiments was conducted to see if there was a requirement that the temporal buffer size match the sequence length for the circuit to be effective. As with the previous simulations the world geography comprises a total of 6 locations. In this case we have three temporal buffers, but the reward sequence only requires that we visit two locations to achieve a primary reward (as is the case in all previous simulations). What we are interested in knowing is whether or not redundancy in the temporal form of the circuit has a derogatory impact on discovering sequence lengths shorter than that required to achieve the reward. The results of this experiment are shown in table 10.3. For comparison purposes we have included the average reward per simulation time step due to random action selection value of 0.06 (this corresponds to 1.0 in our relative fitness measure), and the maximum theoretical average reward value of 0.5 (corresponding to visiting the reward and home locations at alternative time steps).

The three buffer temporal circuit performance is close to that of the two buffer circuit where motor neuron excitation is down-regulated to 0.47 after a reward has been attained. However when we ensure that the excitation level remains unchanged at 0.50 performance is inferior. The reason for the lower average reward attained is due to the creation of loops where a third location is included in a sequence path, where one is not required. This implies that path discovery performance may be optimized when the number of temporal Buffer Units is tuned to the length of the optimal action sequence. This is a negative result, in that such tuning would require either prior knowledge when designing the circuit, or we might need to use some other optimization technique (such as genetic algorithms) to determine the optimal temporal Buffer Depth in a specific case.

Table 10.3: Temporal Circuit Fitness

Circuit	Avg. reward	Rel. Fitness
Random	0.06	1.0
2 Temporal Units (Ex. 0.45)	0.21	3.6
3 Temporal Units (Ex. 0.45)	0.22	3.7
2 Temporal Units (Ex. 0.50)	0.33	5.8
3 Temporal Units (Ex. 0.50)	0.26	4.3
Optimum	0.50	8.3

10.4.2 Three-Layer Temporal Units

Table 10.4 displays the results where a primary reward is only attainable if the agent visits the two asset locations and home location in strict sequence (i.e. the 2,1 and 0 indexed locations in that order without any interruptions). This is different from all experiments up until this point, since previous experiments relied upon the accumulation of assets by the agent in visiting a reward location. The agent in prior simulations could visit a reward location, attain the reward asset, and then visit other locations prior to returning to the home location to attain a reward. However in this experiment, we are only generating a reward signal when the agent visits the 2,1 and 0 indexed locations in strict sequence, without any interruptions.

We also use a three layered temporal buffer storing sensor-motor activation levels for the current state, the prior state, and the previous state before that. Again we use a world geography comprises a total of 6 locations. For comparison purposes we include the average reward per simulation time step for an agent making random action choices 0.004 (shown in the table under the "Random" entry), and the theoretical optimal which corresponds to the agent visiting the locations 2,1 and 0 and then looping back to 2, such that the agent receives a reward once every three time-steps i.e. an average reward of 0.33.

The first experiment involved the approach described in previous simulations where we down-regulate the motor neurons that are implicated in a reward sequence once a primary reward has been detected. The reasoning here is that, unless their inputs are specifically activated, we want other motor choices to be explored (supporting exploration). This approach yields an average reward pay-

Table 10.4: Three-Layer Temporal Units

Circuit	Avg. Reward	Rel. Fitness
Random	0.004	1.0
Down-Regulation (Ex. 0.45)	0.097	23.4
No Regulation (Ex. 0.50)	0.202	48.8
Optimum	0.333	83.3

out of 0.097 approximately 23 times superior to that of random activity. However, problem occurs when the agent randomly selects a location that does not comprise part of the reward sequence immediately after having received a primary reward. This can lead to the agent forming closed loops amongst the locations whose corresponding motor neurons have not had their excitation factors down-regulated, thus reducing the average reward payout. As a further investigation the same temporal circuits were simulated without the down-regulation of excitation. This allows the agent to re-enter the locations which constitute part of the sensor-motor reward chains more easily. The results in this case are significantly improved, resulting in an average fitness payout of 0.202, corresponding to 60 percent of the maximum theoretical payout of 0.333.

10.5 Discussion of Possible Functional Improvements

Before considering the relative merits and de-merits of the alternative circuit forms, it is useful to consider that the delta unit and temporal unit circuit forms have complementary characteristics. The delta unit form is useful where we have delayed rewards. In this case an agent is able to visit a reward location, accumulate the asset, and then travel to intermediate locations before arriving finally at the reward location. We might term such problems *soft sequence* problems, since a terminal reward does not require a strict sequence of action choices to be followed without interruption. The agent is able to visit intermediate states that play no role in the reward payout without invalidating the reward, providing that certain prior states have been visited. Since the delta unit form accumulates historic state data that allows it to determine which state transitions are associated

with a reward sequence, it is robust to discovering the sequence that results in a reward. However, this circuit relies on repeated successful reward sequences to identify which sensor state transitions are associated with the reward attainment. The temporal unit is less well suited to such a task as it guarantees the learning of all actions that immediately precede a reward payout. Given that the final payout can be attained from visiting the reward location from an intermediate node that does no contribute to the reward sequence, the potential to incorporate inappropriate actions into a learned reward sequence is high.

In contrast, we can imagine another class of problems which we might term *hard sequence* problems. A *hard sequence* problem is one which requires that the agent visit a number of states in a strict sequence without interruption. Under such circumstances the one-shot learning of the temporal unit, where all prior actions are incorporated into the learned action sequence is more efficient than the delta unit. This is particularly true where the number of states in the reward sequence is large, and the space of alternative states is also large, since the chances of discovering the reward sequence through random exploration on multiple occasions (as required by the delta unit) is correspondingly low.

We might imagine that there exist complex action selection problems which comprise both *soft sequence* and *hard sequence* sub-sequences, such that a hybrid circuit comprising both delta and temporal Units working in conjunction may be required. Investigating this hypothesis could be a target of future research.

Another issue worthy of future research is that of competition between temporal Units of different buffer depth. The subsection entitled 'redundant temporal unit' (section 10.4.1) discusses an experiment that compares the relative performance of a dual-layered versus triple-layered temporal units. The optimal sequence comprised only two steps. Whilst the three-layer version achieved reasonable results the dual-layered version was superior. In the case of *hard sequence* problems, where we do not know the optimal sequence length in advance, we might imagine that multiple temporal units working in competition might achieve the best results. How to implement such a competitive circuit is another topic for future research, however, one possible solution is to use tonic dopamine levels as some basis for modulating banks of competitive modules. For example, high tonic-levels associated with frequent use might be used as the basis for

down-regulating the excitation of one competitive action selection unity versus another. Thus resulting in a balance between exploration and repetition.

We have not as originally anticipated seen the benefits of excitation down-regulation within the experiments described herein. However, the two research topics identified; firstly that of combining temporal and Delta Units, and secondly that of allowing competition between temporal Units of differing buffer depth could well make use of this feature.

Given the simplicity of the current model, intentionality or control signals intended to switch between different goal tasks is not supported. We previously hypothesized action selection modes that comprise; *selection*, *exploration* and *exclusion*) Of these only *selection* and *exploration* are evident in the model circuits. The lack of a requirement for the down-regulation of motor neuron activity (once it has been identified as part of reward chain) obviates the need for the *exclusion* mode. However these simulations all relate to the attainment of a single goal. If we were to build a more complex simulation that entailed learning two different goal tasks, we might find that an *exclusion* mode (supported by the down-regulation or motor neuron excitation) provided us with a mechanism by which goal switching could be supported.

10.6 Comparable Networks and Alternative Approaches

10.6.1 Key Features of Our Approach

In this chapter 8 we covered a number of topics; a novel neuron model, where signalling is modulated by excitation levels; a circuit example demonstrating signalling adaptation resulting in energy savings; and a hypothetical model of action selection based on the modulation of excitation factors. Distinctive in the presentation of these examples compared with that for more conventional models is that idea that energy efficiency may manifest itself in the functional operation of neural circuits. In this chapter we described variations of the action selection model presented in chapter 8, together with an example of the use of

such neurons. It is worthwhile considering how this approach compares with that taken by others.

The first thing to note is that the approach taken here incorporates a different conceptual approach from that typically involved in devising conventional neural networks. It starts from the axiomatic position that selection for energy efficiency has shaped the evolution of neural networks, and seeks to explore the explanatory power of this axiom. One of the conclusions that results from this is that variations in absolute or average signalling levels between two arbitrarily selected signalling pathways does not necessarily indicate a difference in functional contribution. Adaptation in the receptivity of target neurons affects the input signals required to attain the same outputs (see 8.4). The most significant impact this has on the creation of circuits that incorporate such features is that competition between channels may occur based on the amplitude modulation, irrespective of the semantic value of the information transmitted between these channels. This corresponds to an additional degree of freedom, compared to circuits where the semantic content of a signal relies upon the entire range of values between the intervals of 0 to 1 or -1 to +1

The second key issue relates to how inputs are combined in our circuit models. It is intrinsic to the class of circuits explored in this chapter is the way in which unique inputs are integrated to generate combinatorial intermediate signals. Although we refer to the dynamic generation of inter-neurons, this relates to the abstraction used in implementing such circuits. The underlying idea is that the densely interconnected dendritic structures that inter-leave source and target neurons in biological systems support almost all combinatorial variations in signalling pathways that we can imagine. It is through the co-activation of source and target neurons that we imagine combinatorial signalling pathways emerging. Our dynamically generated 'interneurons' are intended to be representative of these pathways.

These are the two key features that differentiate the reward adaptive circuits that we have explored in this thesis; firstly, amplitude modulation of channels (as distinct from signals), and the additional degree of freedom that this represents in the construction of neural circuits, and secondly intrinsic support for functionality that relies upon adaptive processes that relate to combinatorial inputs.

There are other aspects of the circuits described here, that are more reminiscent of more conventional approaches. We can define a class of *reward adaptive action selection* (RAAS) systems that incorporate the same essential features;

> Any system that incorporates some 'memory' of a context that exists when a beneficial action is selected, and has the potential to detect this same context (through some comparison of memory with current state) when it reoccurs, may might make use of this detection signal to bias action selection in favour of choices that have historically been associated with the attainment of a reward.

This may sound simple, but hidden within this description is one major problem, how do we determine which of the context features are significant, and specifically;

1. does the action selected depend on a combination of sensor signals, and if so which combinations?
2. which of the signals detected in the context is invariant with respect to attaining the reward, and which ones neutral?
3. how do we support partial matches (i.e. is the system subject to graceful degradation, where partial matches might still elicit favourable actions)?
4. how do we discriminate between competitive matches (i.e. what happens where partial matches correspond to conflicting actions)?
5. how do allow for the discovery of new or novel actions, whilst continuing to support 'remembered' actions?

In our approach we address the first issue through the dynamic creation of 'interneurons' each of which represents a specific combination of inputs. However, we do so only *after* a reward has been detected. This avoids the need to represent all input sensor combinations continuously, avoiding an explosion in the memory and computational requirements that might otherwise be necessary. The delta unit formulation (see section 10.2.4) is able, through a biologically plausible analogue of back-propagation, to gradually determine which of the sensor signals are

needed and which are superfluous to action selection, and thus addresses the second issue. The second and third issues are addressed through our winner-takes-all output stage. The last issue is more challenging. Whilst we originally anticipated that the system outlined in section 8.5.1 would support selection, exploration and the exclusion of actions through reward based adaptation that affected neuron excitation levels, we found our results to be somewhat problematic. Our initial approach allowed for the possibility of detrimental loops, whereby down-regulated neurons associated with reward pathways, were potentially excluded from consideration during exploration. In order to rectify this, it was necessary to avoid the down-regulation of such neurons. This is arguably the area that requires the most consideration in future research.

10.6.2 Alternative Approaches Compared

Jordan (Jordan [1986]) and Elman (Elman [1990]) circuits also fall into the class of reward adaptive action selection systems described in the previous section. Essentially the combinatorial problem is addressed by the fully-recurrent connections between the sensor neurons and the hidden layer. The 'context units' provide the system with some basic memory such that prior states also effect current signals. Jordan's variation simply takes the input of the context units to be that of the output layer, rather than the hidden layer. Both network varieties are classified as simple recurrent networks. As mentioned previously our approach is potentially more efficient, since it relies only on the processing of combinations that have already been implicated in the attainment of a reward. Nodes grow gradually, and could be subject to dynamic pruning. Our approach also allows for arbitrary time or sequence periods. Elman circuit performance decreases as sequence length increases (Portegys [2010]). Capi and Doya investigated a number of Elman like variants in a simple robot navigation task (Capi and Doya [2005]). One of the conclusions reached was that an architecture that incorporates competitive memory modules out-performs a standard Elman circuit. This provides a hint as to a possible direction in which our work could be improved, as envisaged in section 10.5.

Whilst the Elman circuit could potentially be used as the basis for some kind

of reinforcement learning system, its primary purpose was simply to demonstrate how state behaviour over time could be incorporated into networks. The MAXON architecture is more closely inspired by reinforcement learning systems (Crabbe and Dyer [2001a] and Crabbe and Dyer [2001b]), and is therefore more directly comparable with the circuits that we have developed. The MAXON architecture is divided into two parts; the *policy* network and the *value* network. In some ways it represents a broader scope than addressed by the circuits presented here. The *policy* network is functionally equivalent to our reward adaptive circuits, in that it selects an action based on current sensor state. What is missing from our formulation is the *value* network, whose function is to allow for the prioritization and disambiguation of multiple goals. If we were to incorporate such features, this would represent an additional 'layer' on top of circuits that we have already presented. Again this ties in the potential use of multiple competitive circuits, with some system for arbitrating between them. Further inspiration for how such functionality might best be implemented might be gained from a more detailed consideration of the interior and exterior sections of the globus pallidus as they relate to the striatum (see section 6.2).

The main reason for the switch in emphasis from the system view of chapter 7 to the neuron and circuit perspective of chapter 8 was the recognition that the design of a reinforcement learning *system*, whilst informative, precludes the use of the evolutionary robotics methodology. To be of value, a reward adaptive circuit suitable for neuroevolution needs to be simple enough that it allows for complexification through fitness dependent selection. However, an understanding of the system perspective certainly should inform the decisions made in the design of these 'seed' circuits. It is from the system level perspective, that we are able to imagine the dimensions that should be explored. System level models of reinforcement learning are quite well developed. We shall compare our model with that of the PVLV model (Hazy et al. [2010]) in section 11.4, however other models, such as those presented by Grossberg (from Grossberg [1992] to Grossberg [2011]), and a model intended to represent the pre-frontal cortex and the striatum (Dominey [1995]), merit further consideration in pursuit of development the models that we have presented in this chapter.

This thesis differs quite significantly from much of the work that has already

been carried out on the topic of reinforcement learning, and associative memory, shaped by rewards in the evolutionary robotics community (Whitley et al. [1993], Yamauchi and Beer [1994c], Niv et al. [2002], Blynel [2003], Igel [2003], Taylor et al. [2006], Whiteson et al. [2007], Izquierdo [2007], Schembri et al. [2007], Soltoggio et al. [2008], Izquierdo et al. [2008]). To a certain extent we have eschewed more conventional models for an approach built on a foundation of energy efficient signalling inspired by biological systems. In doing so we have arrived at a circuit model, that whilst it presents certain interesting features, has yet to be fully explored in an evolutionary context. Another difference, is that of a desire to identify complementary circuits, that when combined might yield considerably more complex behaviour than we have as yet been able to achieve. We are particularly interested in how GasNets and circuits intended to support locomotion might integrate with reward adaptive circuits of the type described here (we shall address this topic in more detail in chapter 12). This is another reason for seeking to devise a circuit of minimal complexity, that might allow for co-evolutionary approaches to the integration of complementary networks. Work carried out by those authors cited earlier, will no doubt inform future work once evolutionary experiments are revisited.

Chapter 11

Biological Comparisons

11.1 Introduction

This thesis has been written from the perspective of an evolutionary robotics researcher. Every effort has been made to arrive at a model of reward adaptive circuits that is in some sense biologically plausible. It is however, only through looking again at biological systems that we are in a position to determine whether or not a claim of biological plausibility is in any way justifiable. It is no doubt the case that the sense in which we use 'biologically plausible' here is less rigorous than the sense in which someone working in the field of computational neuroscience might use it. Despite this, there are certain aspects of the circuit models presented that bear some correspondence with features discernible in the basal ganglia system that we seek to identify in this chapter. Over the longer term, it is hoped that the role of this chapter, is to provide us with some basis for considering how we might extend our existing model. No doubt there are certain limitations of the model that are worthy of further consideration. Our preference, if at all possible, is to identify comparable features in biological systems and try to imagine how such features might provide us with some basis for improving our model, rather than seeking to make arbitrary improvements based on engineering considerations alone (taking us in the direction of machine learning).

11.2 Model comparison with the striatum

From a 'bottom-up' perspective in chapter 8, we considered energy efficiency and possible adaptations that would help us to explain the depressed output of dopamine neurons in response to failed expectations. We arrived at a simple model neuron whose transfer function could be modulated by an excitation factor. We then posited the idea of tri-modal signalling, whereby a competitive arrangement of such neurons would support action selection through three distinct modes; selection, exploration and exclusion.

In work described in chapter 9 and 10 we started with a minimal implementation of a circuit constructed with such neurons, and sought to explore how incremental adaptations might lead to increasingly sophisticated reward anticipatory circuits. In doing so we arrived at two distinct forms comprising Temporal Units and Delta Units, with complementary characteristics. Whilst Temporal Units support the learning of strict sequences, Delta Units support the learning of delayed rewards (where the reward payout is robust to the visitation of certain intermediate neutral states). A surprising result was that down-regulation of motor neuron excitability was not necessary to support the efficient discovery of reward sequences.

This lead us to the conclusion that Exclusion is not required to learn a single task. One hypothesis is that they may underlie the ability to learn *multiple* reward related tasks, through selective inhibition. One reason for continuing to believe that this model may still have some validity lies in the properties of neurons in the striatum;

1. The medium spiny neurons (MSN) exhibit what is termed 'Up' and 'Down' states (Wilson and Kawaguchi [1996]).

2. These Up-Down States are modulated by dopamine signals (Gruber et al. [2003]).

In our original model the neuron switches from an excited state to a down-regulated state on the receipt of a reward signal. To use the terminology of the MSN, the pre-adapted state (normal excitation) corresponds to the 'Up' state and the post-adapted (reduced excitation) corresponds to the 'Down'

state. In our model, a neuron in the 'Down' state is typically unresponsive, until its exact inputs are stimulated, in which case, its output exceeds that of other competitive neurons. In the MSN these characteristics are mirrored exactly. 'Down' state neurons are unresponsive unless their specific cortical inputs are stimulated. In our model, a neuron in its 'Up' state is one whose inputs have yet to be closely associated with a specific reward. In comparison with neurons in the 'Down' state, they are far more likely to fire with non-specific stimulation, supporting the exploration mode. Through analogy, this model provides us with a good functional description of the role of the medium spiny neurons found in the striatum.

Similarities between our original model and the striatum are not limited to the Up-Down states of the MSN. In our model it is a Primary Reward signal that initiates the differentiation of neurons from their initially elevated excitation states to their inhibited state. Similarly in the striatum, it is dopamine signalling that supports the switching between the "Up" and "Down" states of MSNs in the striatum (Gruber et al. [2003]). Work supportive of the tri-state mode switching hypothesis includes studies showing that dopamine denervation of spiny neurons in rat striatum results in enhanced excitatory synaptic transmission (Pang et al. [2001]), consistent with the purported role of reward signalling in reducing the excitation levels of motor neurons.

Further similarities extend as far as the mechanics the neuron model itself. Despite referring to an "Excitation Factor", this value is used to shift the threshold value associated with a Rayleigh distribution that determines the probability of firing. This is analogous to the role of (dopamine) D1 receptor activation that raises the threshold for activation of action potentials (Surmeier et al. [1992]).

A comparison of our model with a more traditional model (Schultz et al. [1994]) idea behind the cortical-Striatal relationship is shown in Figure 11.1 and Figure 11.2.

In our simulations the Exclusion of a single action (supported by excitation down-regulation) is not required for learning a *single* reward sequence efficiently. Despite this, the original Selection, Exploration, Exclusion model appears to conform very well with what we find in the striatum. In the absence of additional experiments we can only speculate as to why this might be. An obvious

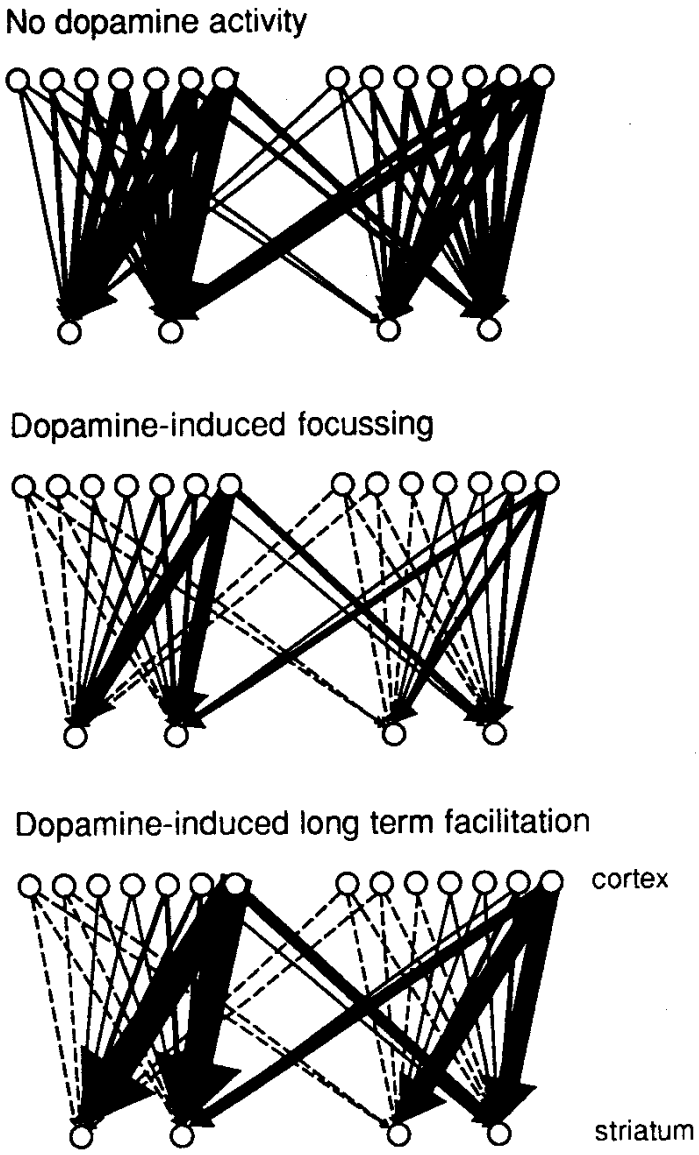


Figure 11.1: **Traditional view of the role of dopamine in the striatum:** (Top) In the absence of dopamine signalling the impact of cortical input to striatal neurons would be minimal, pathways between cortex and striatal neurons have arbitrary weights. (Middle) Initially, a dopamine signal supports an immediate focussing effect, where those inputs which are coincidently active are strengthened.(Bottom) In a hypothetical learning mechanism, dopamine facilitates long-term changes. Arrow width represents the relative synaptic influences on post-synaptic impulse activity by striatal neurons. Source Schultz et al. [1994].

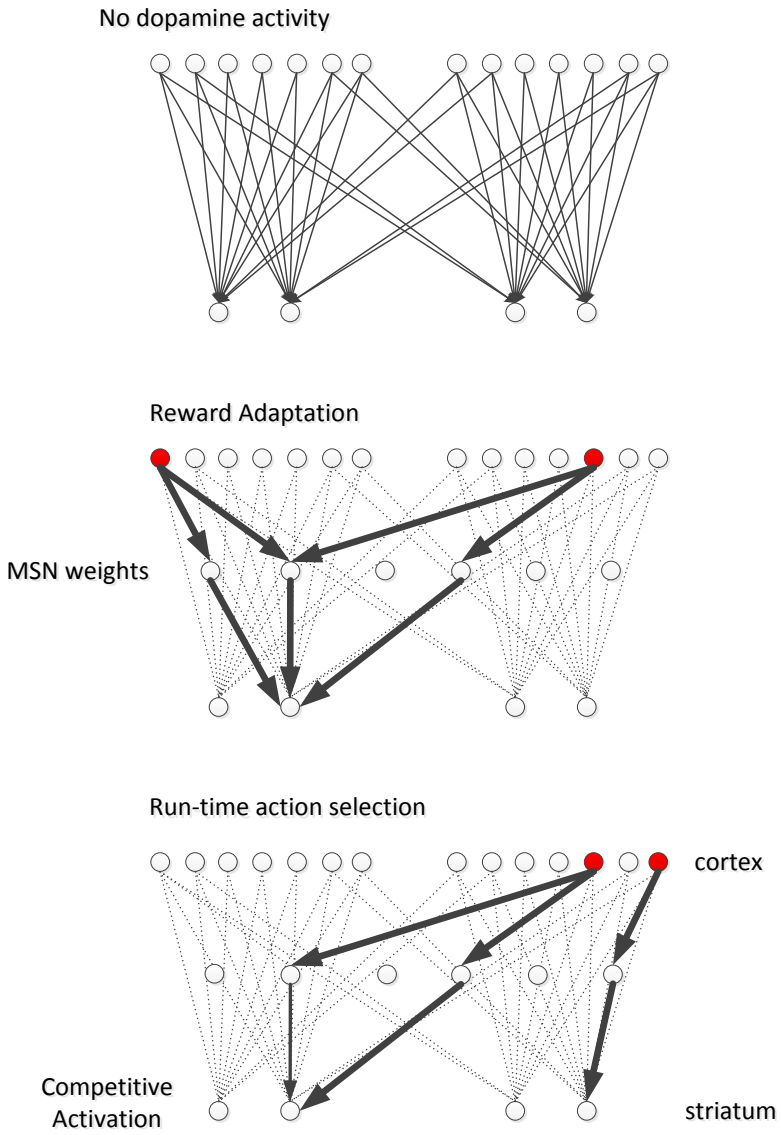


Figure 11.2: **Proposed model of the Role of dopamine in the striatum:** (Top) In the absence of dopamine signalling weak, but extensive connections are made between the cortex and striatal neurons. (Middle) Initially, a dopamine signal supports an immediate focussing effect, where the inputs which are coincidently active are strengthened, to form combinatorial inputs to the active striatal neuron(s).(Bottom) Under circumstances where no immediate reward is detected action selection is determined by the competitive activation of MSN neurons, based on the complete or partial activation of their inputs from the cortex.

hypothesis to propose is that striatal MSN activation is organized in a goal oriented fashion, such that there exists a mechanisms whereby alternative reward chains (supporting multiple goals) are selectively activated. The excitation down-regulation supported by our model neuron provides us with functional support for such a mechanism. In this case Selection, Exploration and the Exclusion concept would relate to support for multiple goals rather than single actions.

11.3 Inferring the functional role of the basal ganglia

The circuits described in this thesis are very simple. Their intended role is to support the learning of action sequences in response to reward signals. These circuits represent the simplest form of reward adaptive circuits that we can imagine, that still provide improved fitness for an artificial agent that may incorporate them. The resultant circuits bear some resemblance to what we find in the striatum. The sensor neurons correspond to cortical inputs, the intermediate neurons to MSN weights, and the motor neurons to the outputs of the MSNs. The winner-takes-all algorithm that we use to chose between motor actions corresponds to the function of the globus palidus. The adaptation that takes place to create new intermediate neurons on the receipt of a reward is analogous to that which takes place in the striatum in response to dopamine phasic activity. If these circuits capture aspects of proto-circuits that may underpin the emergence of the basal ganglia over evolutionary time scales, we may make some inferences regarding the functional role of the basal ganglia by comparison with the attributes of our circuits.

The primitive circuits modelled here are intended to support the learning of action state sequences, which in reinforcement learning would be termed policies. When we refer here to action states, we are referring to action states that resemble those of our robot in Chapter 5. These action states were labelled **Wander**, **Eat**, **Orient** and **Approach**. Each action state comprises a selective input filter on sensor data. A single action state constitutes a mapping of low-level sensor data to the appropriate motor program. Selective High-level sensor data determines

action state transitions and determines the current motor program. Action state sequences, correspond to the state transitions in a finite state machine model. An action state sequence in this context is not intended to refer to the coordination of specific motor sequences (a role that is associated with the cerebellum, rather than the basal ganglia). However, in as much as errors in an action state sequence model would result in inappropriate initiation of motor programs, or potentially corrupt action state sequence transitions, such a model is not inconsistent with that of the role of the basal ganglia (defects of which result in motor control problems in Parkinsons or Hodgekin sufferers).

As far as our model is concerned, usage of the terms *sensor* and *motor* are primarily intended to signify input and output. If we were to apply our low-level model to a robotics application, the inputs and outputs would correspond to sensor inputs and motor hardware control signals. When seeking to look for analogies between our model, and that of the role of the basal ganglia, such terms may well be misleading. However, as our fundamental approach is that of a roboticist, we shall stay with this convention, and seek to apply it as consistently as possible.

Common amongst all the circuits that we investigated is the association between the sensor input and the motor output that is made via the creation of the intermediate neurons. Comparisons with the striatum of the basal ganglia, led us to the conclusion that the functional role of these intermediate neurons is effectively embodied in the pre-synaptic weights of the cortical afferents to the medium spiny neurons. Irrespective of this change in perspective, the fundamental idea encompassed by them is the association of environmental conditions with actions. The purpose of which is to identify action sequences that lead to a reward and ensure that future actions are biased towards repeating these actions under similar sensor conditions to maximize future rewards.

We modified augmented the basic circuit (originally described in Chapter 6) with additional sensor buffers, which gave primitive support to the two action's necessary to arrive at the optimal solution for the specific experimental simulation; that of going to the Reward Location, and immediately returning to the Home Location. We achieved this through two circuit configurations. The first was the *temporal* form, where we stored the sensor and motor states of two simula-

tion time steps, so that we could identify multiple actions that lead to the reward. In the second configuration, the storage for prior sensor-motor states was used as a *delta-unit*. In this latter case, we made associations between the sensor-motor actions that would lead to a positive change in the input most strongly associated with the primary reward. It is this latter circuit form that with appropriately chosen excitation factors, lead to the highest average reward.

There appears to be a significant bias in the amount of research that focuses on the corticostriatal connections, compared to that of the thalamostriatal. A search for the terms corticostriatal (or cortico-striatal) and thalamostriatal (or thalamo-striatal) on the Pubmed website yields 1601 and 167 references respectively. On Googlescholar the results are 23,650 and 3,091 respectively. This indicates that approximately 9.5 (Pubmed) or 7.5 (Googlescholar) more papers are written about the corticostriatal connections than the thalamostriatal connections. Schematics for striatal connections very often leave out the thalamic afferents completely. This bias may appear somewhat confusing, when we consider that the number of thalamostriatal terminals in the striatum is of a similar magnitude to that of the number of corticostriatal terminals (Smith et al. [2004] and Lacey et al. [2005]). Of the work that has looked at both classes of afferents, (see Smith et al. [2004] for a review and Ding et al. [2008] on neuron characteristics) some clear differences have been identified. Whilst it was originally understood that dopaminergic neurons converge with corticostriatal afferents, rather than thalamostriatal afferents, these results have more recently been questioned. Moss et al found that (Moss and Bolam [2008]) found that the spatial proximity of dopamine and thalamostriatal terminals to be comparable with that of dopamine and corticostriatal terminals, arriving at the conclusion that thalamostriatal afferents were just as likely to be influenced by dopaminergic neurons as corticostriatal afferents. In other words, input from the thalamus to the striatum is functionally as significant as that from the cortex.

From our perspective, in seeking to understand the emergence of reward anticipatory circuits, it is also interesting to note that when we compare homologous brain structures, the thalamostriatal connection is more prominent in primitive organisms than the corticostriatal connection (Striedter [2005]). This by itself implies a less significant role for the cortex than that of the thalamus in determining

actions sequences in primitive organisms. From an evolutionary perspective, we might reach the conclusion that it is the thalamus that initially dominated the selection of action sequences. To summarize;

1. The relative number of thalamostriatal terminals compared to corticostriatal terminals implies a significant role for the Thalamus in determining the output of the striatum.

2. Dopminergic neurons may play a similar functional role in the way in which inputs to Striatal Medium Spiny Neurons are processed, irrespective of their origin (Cortex or Thalamus).

3. The pronounced role of the thalamostriatal terminals in primitive organisms might imply a significant role in reactive responses rather than considered actions.

These factors alone seem sufficient for us to consider the importance and functional significance of the thalamostriatal connection to its role of the basal ganglia. Whilst the conventional approach is to see the basal ganglia as comprising a cortico-striatal-nigral-thalamic loop, we might well ask ourselves, whether or not there exists a comparable loop in which the thalamostriatal link is strongly implicated. The most obvious candidate for such a loop is that of the Nauta-Mehler Loop (Kim et al. [1976], Giménez-Amaya et al. [1995], Mengual et al. [1999b]). Lanciego et al (Lanciego et al. [2004]) have identified this as a potentially significant pathway in determining basal ganglia function (this research also relates to the rat). Their studies also show segregated pathways, in a fashion which could be analogous to that which we see in the cortico-striatal-nigral-thalamic pathway. It should be noted, that this particular loop is not generally well researched. As a very rough approximation, this is evident from considering the number of entries found in PubMed for each of the following terms; dopamine (122,663), corticostriatal (1601), thalamostriatal(167), and the Nauta-Mehler Loop(2). Using Google Scholar the corresponding terms return the following number of entries; dopamine (968,000), corticostriatal (23,650), thalamostriatal(3,091), and the Nauta-Mehler Loop(20). It is clear from these figures that the results for Nauta-Mehler Loop

are insignificant in comparison, and probably fair to say that this particular loop is largely ignored in the majority of basal ganglia research.

The figure 11.3 displays a cortex oriented schematic of how we might imagine the basal ganglia to work. In contrast and figure 11.4 shows a thalamic oriented schematic. In this case, the reward sequences that are subject to undergo reinforcement learning are motor actions themselves. Effectively the thalamus routes back the motor action states associated with an action selection for the purposes of determining future motor actions. Rather than focussing on the association that is made between sensor state and motor commands, in the attainment of a Primary Reward, we can easily see how a simple sequence of motor associations, could be made through essentially the mechanism (i.e. adaptation mediated by dopaminergic signalling). In a cortex oriented model, we would associate a given action with a particular set of environmental circumstances, in thalamus oriented model we simply adapt the motor sequence associations to Primary Rewards. Figure 11.5 shows an anatomical diagram of the principal pathways associated with association learning in each case.

A combined cortical and thalamic perspective integrates both environmental (in addition to proprioceptive and higher order associative patterns) together with prior motor action. Such a perspective is different from the conventional perspective that sees the cortex as the primary input to the striatum. Note that in our simple simulations we had two sets of sensor data; endowments and location. Given that our motor choices have a one to one correspondence with the locations themselves, we can equivalently interpret our location sensors as prior motor action selections, in a fashion that is directly analogous to the proposed role that the thalamo-cortical projection plays.

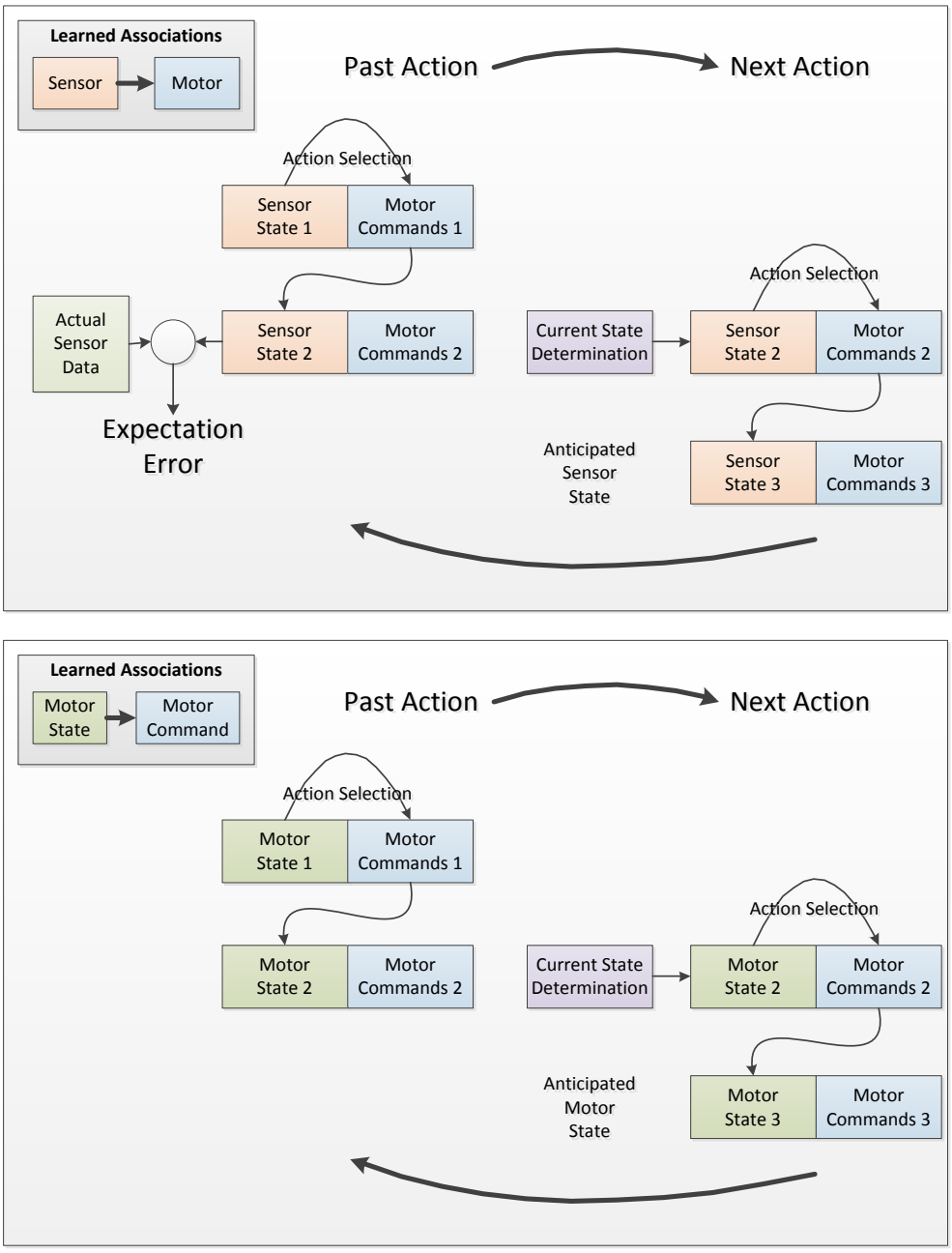
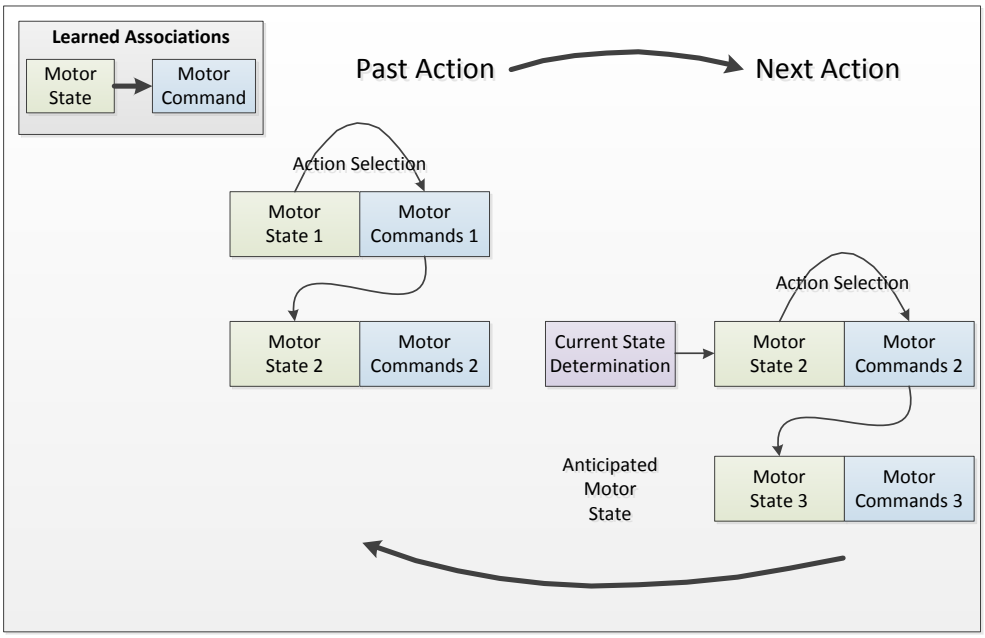


Figure 11.3: **sensor-action vs action-action associations: (Top)** So far we have focussed on "sensor-motor" associations, as the basic building block of *automatic* action Selection. Simply put, a current sensor state, determines the next selected motor action. **(Bottom)** A more primitive framework for *automatic* action Selection could be based simply on the association of motor actions, which lead up to the presentation of a reward.



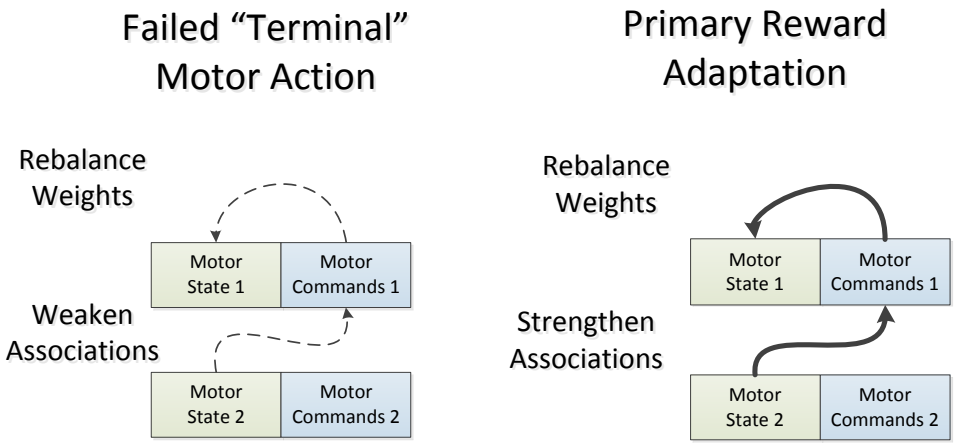


Figure 11.4: **action-action association adaptation: (Top)** This schematic shows how *automatic* action Selection could be made through the association of motor sequences. **(Bottom)** In the case where the agent experiences a Primary Reward, we are able to adapt these motor sequence associations, strengthening those that have lead to the Reward. Unlike in the case of sensor-motor Associations, there is no obvious analogy to re-adaptation based on anticipated sensor states, however we could still adapt the associations based on the final success or failure of the terminal motor action.

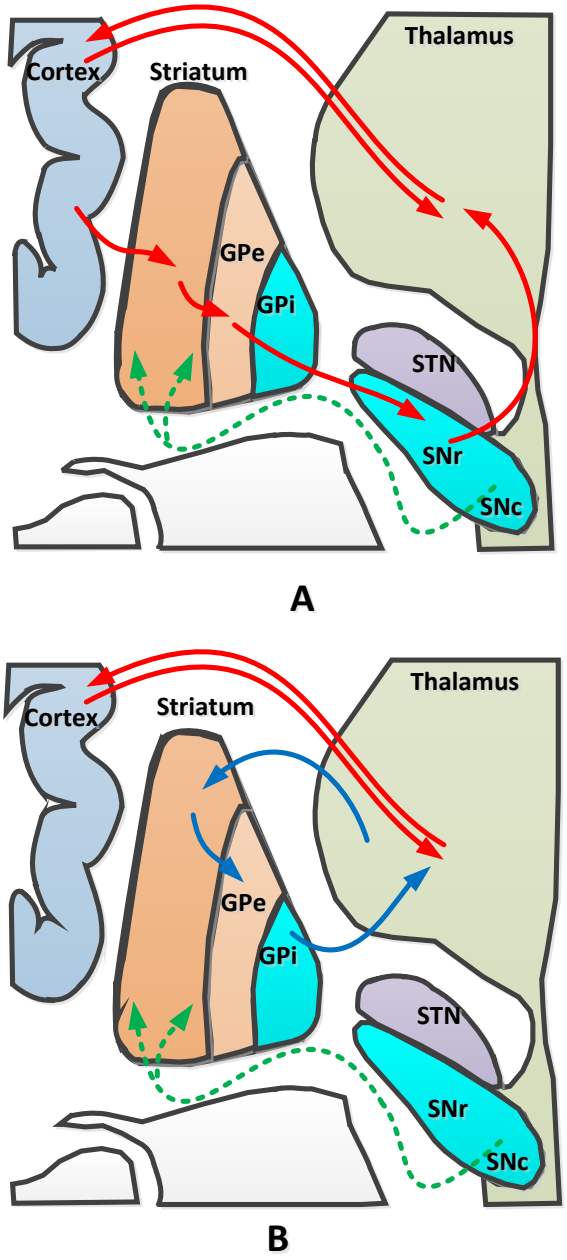


Figure 11.5: **Hypothetical sensor-action and action-action loops: (A)** The top diagram corresponds to our original model where the focus was "sensor-motor" associations. **(B)** At the centre of the diagram we can see the **Nauta-Mehle Loop**. Hypothetically this loop could support the learning of motor sequences via associations between current and subsequent motor patterns. In both cases the dotted line represents the dopaminergic Neurons originating in the SNc. Abbreviations: GPe, globus palidus, external segment; GPi, globus palidus, internal segment; SNr, substantia nigra, pars reticula; SNc, substantia nigra, pars compacta; STN, subthalamic nucleus. Adapted from Wilson [2004].

11.4 Comparision with the PVLV Model

The PVLV model (Hazy et al. [2010]) is perhaps one of the most developed computational models of the brain circuits that support reward anticipation circuits in the brain. One of the difficulties in comparing models is that the basal ganglia supports multiple loops (Garret E.Alexander, Mahlon R. DeLong [1986], Garret E.Alexander, Mahlon R. DeLong [1986], Haber and Calzavara [2009]), comprising parallel circuits that implicate, Limbic, motor, sensor and Associative pathways. Whilst the focus of our model is on the basal ganglia and Largely the motor pathway, the PVLV model takes in the limbic system, and through the hippocampus, potentially also incorporates some aspects of sensor and Associative pathways. Given these limitations, where appropriate we will make comparisons,

The basic idea behind the PVLV model is that the basal ganglia system can be broadly modelled as two complementary sub-systems; the first is a Primary Value (PV) system, which governs how dopamine neurons respond to a reward (a primary reward in our terminology), and a Learned Value (LV) system, which governs how dopamine cells respond to an conditioned stimulus (an implied reward). Its key results or hypothesis are listed below in the form of quotations, followed by annotations and comparisons made by the author of this thesis.

Abbreviations used in the synopses of their hypotheses correspond to; conditioned stimulus (CS), unconditioned stimulus (US),ventral striatum (VS), dopamine neurons (DA) and central nucleus of the amygdala (CNA).

1. "The lateral hypothalamic area is the main site for the reactive representation of US value for rewarding stimuli such as food, water, etc., and this provides the main excitatory signal driving phasic dopamine bursting after primary reward onset."(p.707, Hazy et al. [2010])

   Here the assertion that the lateral hypothalamic area (LTA) is the origin of the signal that we shall refer to as the"Primary Reward" signal is made. This is not addressed by our model, but we are entirely happy to accept that this is the case. Incorporating this feature does not change any of the aspects of our model.

2. "Patch-like GABAergic neurons in the ventral striatum are the main substrate for the learned representation of a US expectation (PVi). Inhibitory projections therefrom to midbrain DA cells shunt excitatory inputs thereby eliminating the phasic burst for the US. The VS also projects via the pallidum to the lateral habenular (LH) nucleus of the epithalamus, helping that substrate to compute when an expected reward has been omitted."(p.707, Hazy et al. [2010])

   The assertion here is that the striatum exhibits adaptation that in some sense encodes anticipation of a "Primary Reward" (note that the "Primary Reward" signal is an Unconditioned Stimulus signal, they are not referring here to anticipated Conditioned Stimulus). The role of detection of a failed reward is assumed by cells in the Thalamus (specifically the lateral habenular nucleus of the epithalamus). Projections from the striatum that relate to the anticipation of a Primary Reward inhibit the phasic response of the dopamine Neurons. This is entirely consistent with our model, but expands on the source of expected reward omission and identifies it with the lateral habenular (LH) nucleus of the epithalamus.

3. "Multi-modal glutamatergic projection neurons of the medial segment of the CNA are the substrate for the acquired representation of CS reward value for the purpose of driving phasic dopamine bursting at CS-onset."(p.709, Hazy et al. [2010])

   Here the authors propose that the Central Nucleus of the amygdala is the source of signals corresponding to the value of a stimulus corresponding to a Conditioned Stimulus (an *implied* reward stimulus in our terminology). Whilst this may be true of the model that they propose, it may not necessarily be the case in our model. Incorporation of the amygdala may imply some involvement in aversive as well as rewarded activity. Similarly the close association of the amygdala with the hippocampus may also bring in the involvement of the Limbic and Associative systems. In our model, the "value" conditioned stimulus is not considered. It is through activation competition in our analogy of the MSN neurons that action selection is

determined, without consideration to value.

4. "The critical learning in CNA occurs at synapses between incoming CS sensory representations and multi-modal cells initially responsive to US, and these are trained by phasic DA at the time of US-onset only. CS triggered phasic DA signals do not train associations in the CNA."(p.709, Hazy et al. [2010])

   Here the authors assert that the learning (that takes place in the central nucleus of the amygdala) of the value of conditioned stimulus (*implied* rewards), takes place only at the onset of the conditioned stimulus (the *primary* reward). In our view, although the location at which learning takes place may vary (as it may be loop specific), our model also ensures that adaptation only takes place in response to the presentation of the Primary Reward. In this respect the two models are analogous.

5. "With overtraining, phasic DA firing to CS-onset is significantly reduced, but persists indefinitely. However, if there is a CS2 prior to, and predictive of, CS1, DA firing to CS1 is eliminated. These effects can be explained in terms of a CS activated inhibitory representation (LVi) that inhibits the excitatory influence from the LVe (CNA)."(p.710, Hazy et al. [2010])

   This corresponds to the widely observed migration of the dopamine activity to the earliest predictive stimulus, i.e. the earliest Conditioned Stimulus ("Implied Reward"). Suppression of dopamine activity to later Conditioned Stimuli is suppressed. This is consistent with out model.

6. "DA firing to CS and US inputs is invariably in the form of a phasic burst, even when these input signals persist for sustained periods of time. The pedunculopontine tegmental nucleus (PPT) or the midbrain DA nuclei themselves appear likely to be responsible for producing this bursting property, in a manner consistent with the temporal derivative Y mechanism now used in PVLV."(p.710, Hazy et al. [2010])

The idea here is that adaptation in the system due to dopamine activity is primarily in the form of a phasic burst, irrespective of the persistence of the unconditioned stimulus (*primary reward*) and Conditioned Stimulus (*implied reward*). This is not an issue that is addressed in our simplified model, but there is nothing to suggest that such a view would be inconsistent with our model.

In conclusion, it is reasonable to say that whilst the focus of attention of the PVLV model is different from ours (with regards to the specific loop that is being addressed), there are no substantial contradictions between the two. One important area of agreement is in system adaptation *only* to the Primary reward presentation, rather than presentation of Implied Rewards (or CS in their terminology). This is consistent with the conceptual schematic shown in figure 8.1 in Chapter 6.

Chapter 12

The Adaptive Replicator Model

12.1 Overview

In this chapter we seek to fully express our notion of an **adaptive replicator**. We first referred to a bauplan for an adaptive replicator in section 2.5.1 as a template for the various processes encompassed by the evolutionary robotics methodology in the generation of agents capable of exhibiting adaptive behaviour. Whilst this bauplan provides us with a framework for understanding how genetic, ontogenetic and continuous processes of adaptation are related, it does not specify the functional details of its constituent parts. In this chapter we focus on one particular level, that of the processes of *continuous adaptation*. These processes support change in an agent, in a manner that is highly responsive to variations in environmental and internal state. Inspired by Ashby's model of ultrastability (see section 2.5.2), we propose an archetype for adaptive organisms. It represents a hypothesis of the functional organization present in the neural circuity of agents capable of exhibiting complex adaptive behaviour. The purpose of this chapter is to present this model, use it as the basis for identifying complementary functional circuits, and in the process justify the choice of GasNets and reward adaptive circuits as the focus of research.

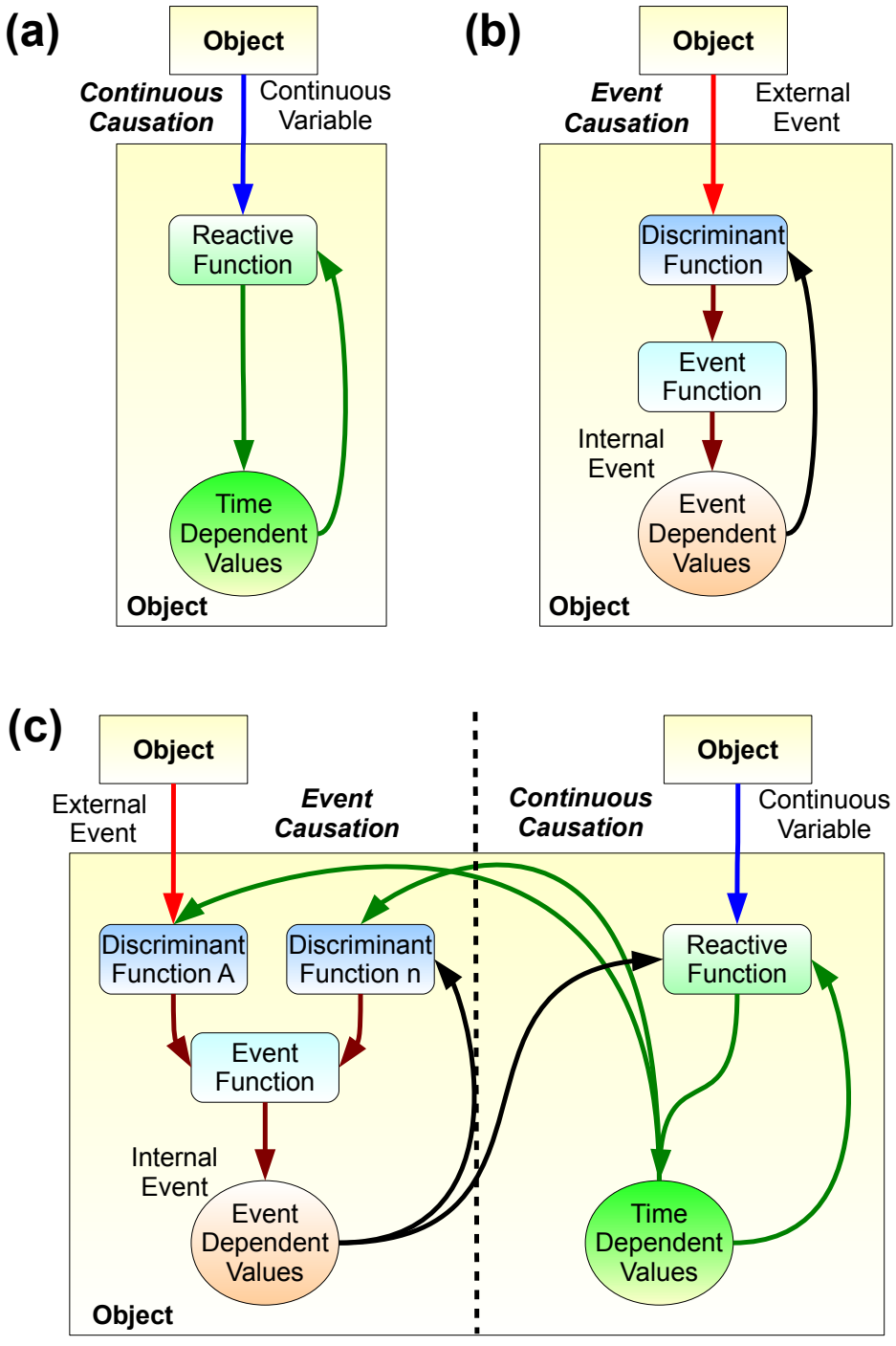


Figure 12.1: **Causation and Adaptivity:** **(a)** One of the least adaptive objects is one whose internal reactive function remains constant. **(b)** An object whose activity is contingent on complex internal and external states is capable of exhibiting more complex behaviour. **(c)** An object that processes complex contingencies and modifies the functional form and parameters of embedded reactive functions is capable of even more complex behaviour.

12.2 Causation and Adaptation

12.2.1 Causation and AI

The seminal text on causation as it relates to AI was originally published by Judea Pearl in 2000. In this book he seeks to describe a formal framework by which causal relationships might be inferred, providing an alternative approach to a purely probabilistic argument, buy incorporating notions of causal structure through graphical representations. Pearl's goal is to seek to answer two key questions;

> How should a robot acquire causal information through interaction with its environment? How should a robot process causal information received from its creator-programmer? (p.413, Pearl [2009])

The second of these questions is of little relevance to the evolutionary robotics methodology, since we seek to generate entities that are able to solve tasks through adaptive processes, rather than those that require programming. Whilst the former of these questions is clearly of general relevance to those in robotics, it is not a problem with which we are primarily concerned here. Our key concern is in seeking to develop a conceptual structure that differentiates between different *kinds* of causal relationships, and how this might relate to some functional differentiation internally within adaptive systems.

Pearl's perspective is typical of classical AI. We might imagine the construction of an artificial brain that enables a robotic to infer causal structure, and through this, build internal representations of this structure. This might then be subject to some symbolic manipulation in which this causal structure acts as constraint on problems that are solved by the robot in the pursuit of certain goals. In contrast, our approach merely acknowledges that causal structure exists (with the world as its best model) and focusses on the functional systems by which changes in the world elicit changes in behaviour.

12.2.2 Causation and Adaptive Processes

The causal structure of the physical world determines the extent to which arbitrary changes might *affect* the organism. It also underlies and the *effect* of actions or activities conducted by the organism. This causal structure is intrinsic to the coupling that exists between the agent and its environment. From an alternative evolutionary perspective, *adaptive processes* are those functional structures that support the survival and reproduction of an organism in response to continuous change and events both in the environment and internally. It does not seem unreasonable to conclude therefore, that these adaptive processes are in some sense *tuned* to the causal structure of the world.

In an attempt to understand how adaptive processes have evolved to take into account this underlying structure in the physical world, we seek to differentiate between two classes of causality; *continuous causation* and *event causation*. This is not intended to imply that two independent causal processes exist in any physical sense. Ultimately, *event causation* is a macro-representation of complex continuous processes. However, there are some benefits to be gained from employing dual representations of causality, where the two alternative representations correspond to abstractions at differing levels. It allows us to present a single model that integrates the two, and through this, explain functional organization that encompasses both of these levels.

In doing so we seeking to define a functional archetype adopting a level of abstraction comparable with that employed by Ashby. It is not intended to be literal, but to capture some general principles that might help us to understand the functional workings of a hypothetical brain. In the preface for 'Design for a Brain', Ashby states that the book "considers mechanisms that change continuously" (page vii, Ashby [1960]). Ashby's model of ultrastability comprises a dual-loop architecture. The first loop controls reactive responses to changes in the environment. The second loop modifies *parameters* of the first loop (the *reacting part*) when *essential variables* transgress beyond certain bounds. Despite Ashby's initial reference to continuous processes, step-functions play a key to his model; "The step mechanisms affect the reacting part; by acting as parameters to it they determine how it shall react to the environment" (page 98, Ashby [1960]).

Our model is in some ways an analogue of that proposed by Ashby. Where we differ is in our focus on causation as an explanatory principle and in the definition of internal functional structure using a set-theoretic approach (for an extensive exposition of the potential role of category theory in neuroscience see Ramrez [2010]). From our perspective, the *first loop* is primarily concerned with processes of *continuous causation* and the *second loop* is concerned with *event causation*. By augmenting our set-theoretic model with ideas that relate this to causality, we illustrate how the internal functional structure of our model relates to behaviour in the real world.

Note that typically causation is considered with respect to passive objects. In this case, gravity acting on a falling apple might be considered the cause of apple's acceleration (an example of continuous causation), and a hammer striking a cup might be considered the cause of it smashing (an example of event causation). We wish to extend this idea to active objects. When we refer to continuous causation, we imagine as an iconic example, the activity engaged by a man in placing a coffee cup on a table. When we refer to event causation, we imagine the response of a rabbit to the sound of gunshot. Intuitively we are able to recognize significant qualitative differences between the two. In the first case, the man causes the cup to move through the dynamic control of agonistic an antagonistic muscles groups (an example of continuous causation). In the second case the sound of the gunshot represents a discrete event, of some significance to the rabbit, that results in it escaping (an example of event causation). An attempt is made to capture these qualitative insights through defining certain functional structures.

We start from the idea that the internal functional structure of an object determines its *adaptive potential*. Figure 12.1 illustrates schematically three examples of objects that possess varying degrees of adaptive potential. A *simple object* is one that embodies a fixed function, whose output varies with environmental state; figure 12.1 (a). An object whose activity is contingent on complex internal and external states has the potential to exhibit more complex behaviour; figure 12.1 (b). A *reactive object* that processes complex contingencies and modifies the functional form and parameters of embedded functions is capable of even more complex behaviour; figure 12.1 (c). We could imagine an object of arbitrary complexity with an even greater capacity for adaptation. However, our goal is to

seek to determine some general principles of functional organization that allow us to envisage how sub-systems may be organized, integrated, and specialization might take place. Ideally we wish to identify some *canonical forms* that exhibit varying degrees of complexity and adaptive potential. Not only does this given an insight into the functional integration of neural sub-systems, it also provides us with clues as to how such systems might be created through approaches that employ incremental evolution.

We arrive at the model of an *adaptive replicator* through incremental steps. We first define what we mean by a *reactive object*. This is an abstract entity capable of modifying its activity in response to continuously varying values or in response to external events. We then define an *adaptive object* as a *reactive object* that has the additional ability to modify its behaviour in response to rewards. Finally we define an *adaptive replicator* as an variety of *adaptive object* that is produced via the evolutionary robotics methodology (itself an abstraction of biological evolution). The conjunction of these two perspectives results in the model of a notional organism, comprising both; functional elements (that imply the internal workings of the organism) and procedural elements (comprising genetic, ontogenetic and continuous adaptive processes) that indicate how such systems may be generated.

12.2.3 Continuous Causation

An example of continuous causation would be a man holding a cup by its handle, thereby 'causing' the cup not to fall. We might imagine in such a case, that there exists a control system that functions to maintain the cup in a stable position. Such a system might control agonistic and antagonistic muscles to create a rigid structure that allows for the support of the cup. The processes that support such activity are engaged in what we might term continuous causation, and their effect can be modelled by continuous functions with a minor change in notation intended to reflect the existence of a causal relationship.

12.2.3.1 Functions and Causation

When we see the words function and causation written together there is a certain jarring. The function is essentially a notion, whereas the roots of ideas of causation originate in philosophy. It is unclear how we relate the two. We have a strong intuitive sense of what we mean by *causation*. In its simplest form, it comprises the following two elements;

- The change in object (P) is *sufficient* to elicit a change in object (Q).
- The change in object (Q) occurs after that in object (P).

Intuitively, if we apply a force to an object, then we would expect the object to accelerate. This is typically expressed in the functional relationship;

$$F = ma \tag{12.1}$$

Where F corresponds to force, m to the mass of the object and a to acceleration. Alternatively we might write this relationship in a form where v represents the velocity of the object.

$$\frac{dv}{dt} = F/m \tag{12.2}$$

In this case the we are referring to the infinitesimal change in velocity that takes place over an infinitesimal period of time. In the domain of causation, we acknowledge that the application of a force *causes* the acceleration of the object, and implicit in this idea is that the force is applied prior to the acceleration taking place. However, in the functional domain there is no recognition of this temporal precedence. An alternative form, that might make explicit such a precedence is rarely used;

$$\frac{v_1 - v_0}{t_1 - t_0} = F_0/m \tag{12.3}$$

Such a formulation, whilst it implies that the application of the force precedes the change in the velocity of the object, it loses the sense in which a continuously applied force results in a continuous change in velocity. Equation 12.2 is

an elegant shorthand for the mapping that exists between the set of values corresponding to the velocity of an object at t to the codmain of values corresponding to the velocity of the object an infinitesimal period afterwards under the application of a force F that captures the continuous effect of this force. The relationship that exists between the velocity of the object and that of the applied force is one that we shall term **continuous causation**. Even though the functional description of this relationship does not make the precedence relationship that exists between the two explicit (i.e. the requirement for the force to be applied *prior* to the acceleration taking place), we cannot deny that it represents a functional description of this causal relationship.

12.2.3.2 A Notation for Continuous Causation

Let us consider a general function of the form;

$$y = f(x) \tag{12.4}$$

The explicit meaning of this equation is that there exists a domain corresponding to a set of values, an element of which is represented by the dummy variable x, that can be mapped via a function f to a codomain whose values are represented by the dummy variable y. Sometimes we might refer to x as the independent variable and y as the dependent variable, but even here we are not explicitly defining a causal relationship, although the existence of such a function might imply that a causal relationship might exist. In the context of our prior discussion on causation, the meaning of this equation is now ambiguous. Whilst it might describe the dynamics of a causal relationship (as in equation 12.2), the functional description does not tell us if this is the case. If g is the inverse of function f then we can just as equally write;

$$x = g(y) \tag{12.5}$$

We remain uninformed as to the existence of any causal relationship that might exist between the values of x and y. If we were to include such information then we require an alternative notation, such as := where the colon side indicates the dependent value. We are then able to rewrite the equations 12.4 and 12.5

where x is a causal value and y the dependent value in this form;

$$x =: g(y) \text{ or } y := f(x) \tag{12.6}$$

Returning to 12.2, we are then able to express causal relationships in the following fashion;

$$\frac{dv}{dt} := F/m \tag{12.7}$$

$$\frac{dv}{dt} =: F/m \tag{12.8}$$

Equation 12.7 might then refer to the application of a force (the cause) to an object such that he object moves (the effect), equation 12.8 might refer to the application of a force to an stationary object (the effect) due to the collision of a moving object with it (the cause). The use of := notation is inspired by the assignment operator used in some computer languages. For example in Pascal, the following line of computer code results in the assignment of the value of $x+1$ to the variable x.

$$x := x + 1 \tag{12.9}$$

We shall adopt this notation to imply the existence of a causal relationship. In this case := can be read as *is caused by* and =: can be read as *causes*.

12.2.3.3 Adaptation and Continuous Causation

When we consider adaptation we are concerned with changes that take place, that are to the benefit of the entity that embodies these adaptive processes. If we are to consider processes of continuous causation, the simplest causal relationship may be represented by;

$$y := f(x) \tag{12.10}$$

This is read as 'y is caused by the function f of x'. There are two ways in which we might imagine that a this causal relationship might be adaptive. The

first is through parameter modification, of the form;

$$y := f(x, w) \tag{12.11}$$

The second is through functional modification of the form;

$$y := f(x) \quad \text{where } f = g[u] \tag{12.12}$$

In this case we are employing $g[]$ to represent a functor in the sense that it is used in computer programming. By varying an index u, we are able to select a function f from a finite set of functions that take the same parameters. By combining the two we arrive at;

$$y := f(x, w) \quad \text{where } f = g[u] \tag{12.13}$$

Equation 12.13 then describes a process of continuous causation, where variations in w and u support parametric and functional adaptation respectively.

12.2.4 Event Causation

Intuitively we have a strong grasps of event causation. If a lighting bolt were to strike a golfer, thus killing him, we would have no hesitation in saying that the lighting bolt caused the death of the golfer. Although there may be some continuous processes that underlie this event, we still see it as discrete event. When we stop at a traffic light whilst driving a car, we wait for the lights to change from red to green before proceeding. If a driver recommences their journey when the light changes from red to green, we might also say that this event (the light change) caused the driver to continue. However, there is a sense in which this latter example is different from that of the former. We are conscious of the fact that if the driver were waiting to pick someone up, then until this person arrived, they would not proceed, irrespective of the colour of the traffic lights. The light changing might not be sufficient to cause the driver to continue on their way.

There are clearly differing degrees of event causation. We are able to differentiate between them through the notions of *sufficient*, *necessary* and *contributory*

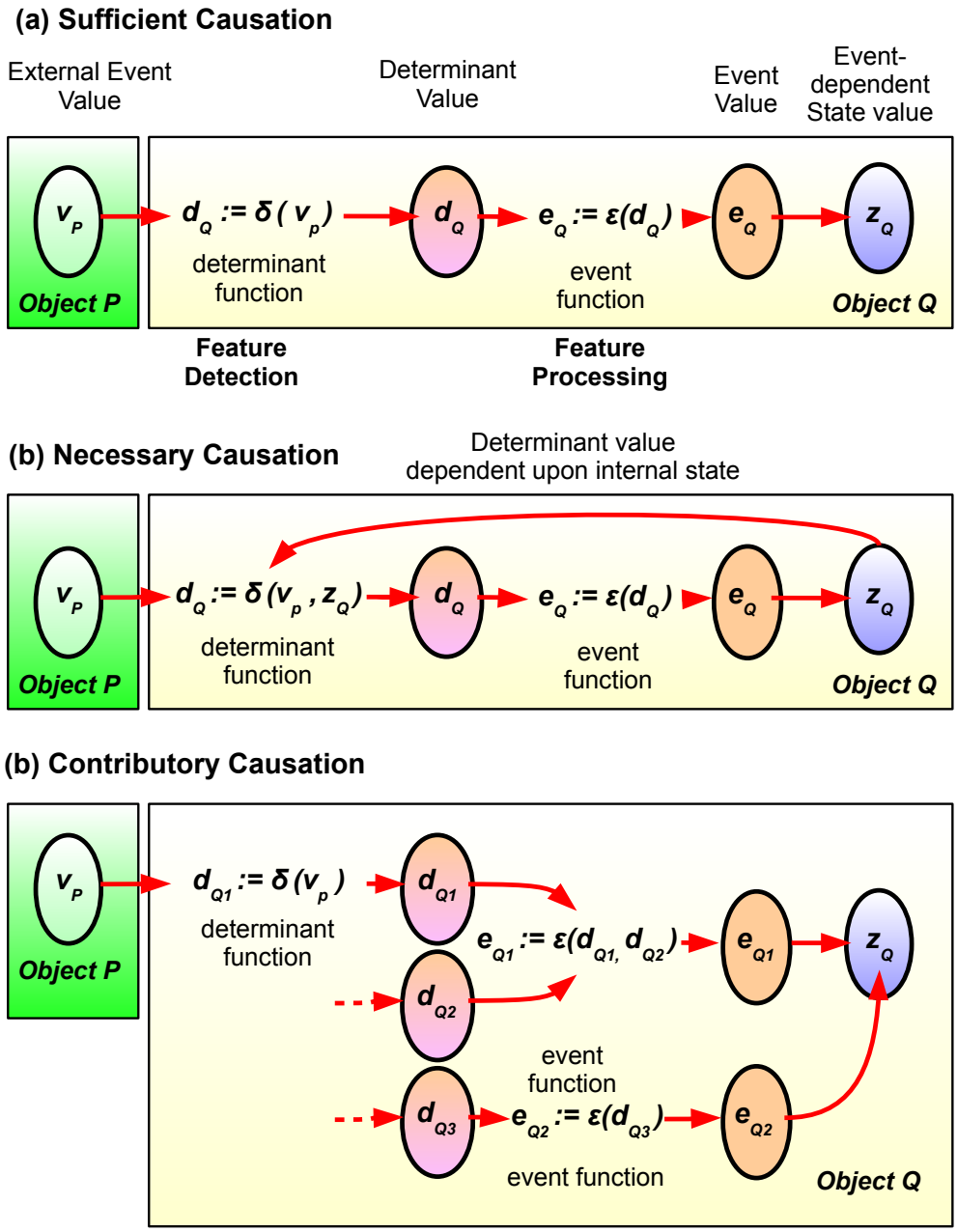


Figure 12.2: **Notions of causation: (a) Sufficient causation** is where a change in the value of v_P always causes a change in the value z_Q. **(b) Necessary Causation** is where a change in the value of v_P is necessary to change the value z_Q (i.e. it will not change without it), but is not guaranteed to do so. An example of such a case where it does not do so, is where the change itself is reliant upon some other internal state value.**(c) Contributory Causation** is where a change in the value of v_P is not necessary to cause a change in the value of z_Q (i.e. there is some independent process that might cause this change), and even if there is a change in v_P it does not to guarantee that a change in z_Q will occur.

causation. Before discussing these however, it is worthwhile point out additional point. When the lights change and the car moves, is it the lights changing that cause the car to move, or the decision made by the driver to release the breaks and press on the accelerator? When talking about causation it is useful to identify certain objects, that encompass properties, that have at any given point in time, certain values. It is then through relating dependencies between values possessed by these various objects, that we are able to more clearly describe causal relationships. In this case the traffic lights might correspond to one object, the driver and car other objects. The property of illumination associated with the traffic lights object, and their corresponding values (on or off) undergo changes that elicit a cognitive responses on the driver object, that might cause them to engage in certain actions.

Returning to the notions of *sufficient*, *necessary* and *contributory* causation, let us define what we mean. Imagine that there are two objects P and Q, and that P has a certain property that has the value v_P. Object Q also possesses a property whose value is given by z_Q. Under these circumstances *sufficient causation* is where a change in the value of v_P always causes a change in the value z_Q. *Necessary causation* is where a change in the value of v_P is necessary to change the value z_Q (i.e. it will not change without it), but z_Q is not guaranteed change even if v_P does so. An example of such a case might be where the change itself is reliant upon some internal state value of object Q. *Contributory causation* is where a change in the value of v_P is not necessary to cause a change in the value of z_Q (i.e. there is some independent process that might cause this change), and even if there is a change in v_P it does not to guarantee that a change in z_Q will occur. These definitions apply just as equally to continuous causation as they do to event causation.

As mentioned earlier, our focus is on the functional structure that might underlie such manifestations of causality. Figure 12.2 helps to illustrate possible functional structures that would support the observed effects. When considering have events are processed by the brain, we might imagine two classes of functions. The first, which we shall term causal **determinant functions** and denote by the symbol δ act as feature detectors. The second, which we shall term causal **event functions** and denote by the symbol ε are responsible for the complex processing

and mediation of detected features for the purposes of triggering internal changes. It is the output of these event functions that associate with the idea of **internal events**.

Whilst the terms *sufficient*, *necessary* and *contributory* causation are useful for describing different varieties of causation, it is clear that a number of alternative functional structures might underlie the processes of *necessary* and *contributory* causation. They remain useful only from a very general perspective of illustrating how different degrees or varieties of causality might manifest themselves. We therefore concentrate on the functional structures themselves to illustrate how systems that process events might relate to those of continuous variables. In doing so we soon find that we are in need of quite a large repertoire of symbols to explain the relationship between differing parts of the system. For this purpose we are forced to introduce a new nomenclature before continuing with our exposition.

12.2.5 A Nomentclature for Reactive and Adaptive Objects

Figure 12.3 displays the nomenclature for object properties and their corresponding values. We shall first explain these terms before going on to build a model that employs them. Generally we shall use upper-case characters to denote sets and lower-case characters to denote an element of a set. There is one exception to this rule; that is in the use of the letters P and Q. We shall use these upper case characters exclusively for representing *objects*. We shall use lower-case Greek letters to denote functions and functors, and upper-case Greek letters to denote sets of functions and functors.

12.2.5.1 Properties and Values

We start with an abstract set of *objects*. One such object might be a cup, for example. To differentiate this cup from any other cup, we will assign it a letter, unique to this object P. This object has a number of *properties*. The set of properties for the *object* P is A_p. The cup might have a 'number of handles' property a_p. If we wish to refer to another property, the 'colour' of the cup, we need a way of differentiating between them. For this purpose we use an index

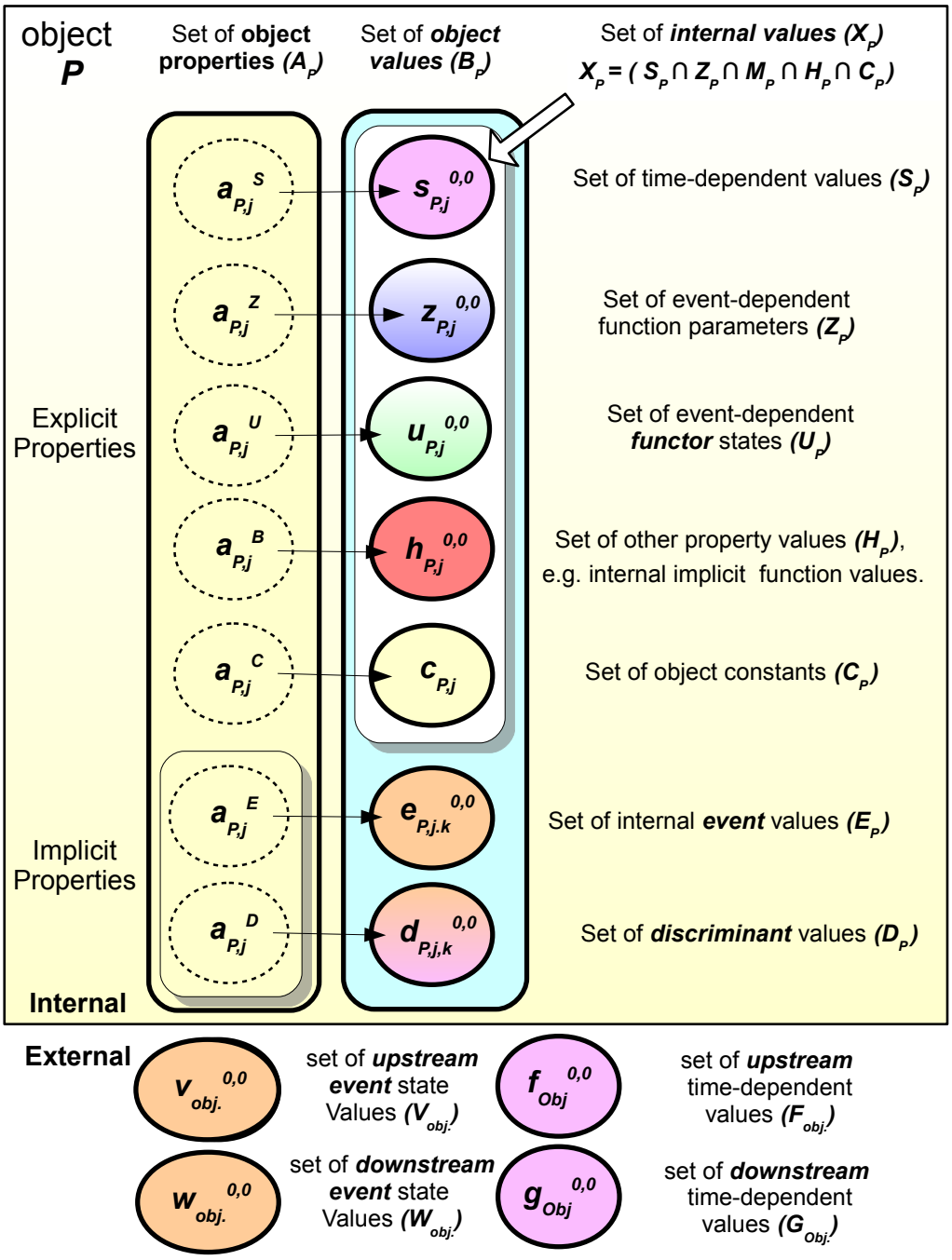


Figure 12.3: Nomenclature for Object Properties and their Corresponding Values

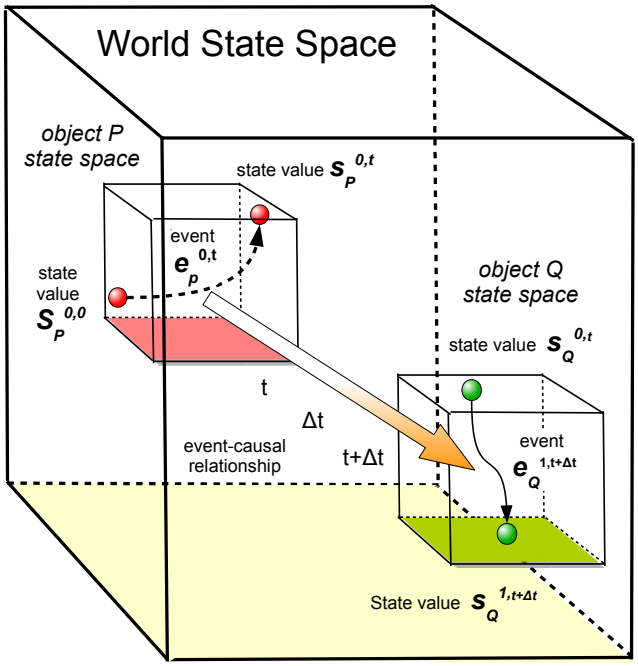


Figure 12.4: An Abstract Event

variable. The 'number of handles' property is then $a_{P,0}$. and the 'colour' property is $a_{P,1}$. This allows us to refer to an arbitrary property by the term $a_{P,j}$, where j is an index variable.

In addition to the set of properties A_p, the object P also has a set of values B_p. For each element in the set A_p there exists a corresponding element in the set B_p that represents the *value* of the property. For example, the *value* of the 'number of handles' property $a_{P,0}$ is 1, the *value* of the 'colour' property $a_{P,1}$ is pink.

Now let us introduce a third property, corresponding to the 'age' of the cup. Unlike the first two properties this property changes with time. So as to differentiate between these two property types, we assign each property to subsets containing elements of the same property type. The first subset A_P^C corresponds to the set of constant properties, and the second subset A_P^S corresponds to the set of properties that are time-dependent. The elements of the set A_P^C are now $a_{P,0}^C$ and $a_{P,1}^C$ for the 'number of handles' and 'colour' properties respectively. The

subset A_P^C only has one element $a_{P,0}^S$, the 'age' of the cup. For reasons of convenience we shall use $a_{P,j}^C$ and $a_{P,j}^S$ to refer to arbitrary elements in each of the sets A_P^C and A_P^S, rather than $a_{P,j}^C$ and $a_{P,i}^C$. Strictly speaking we should us both i and j as indeces to indicate that we are referencing elements in different sets. For convenience however, j is to be interpeted as a local index, specific to the property subset, rather than an index that enumerates all properties in the set A_p.

As mentioned previously, each property of object P has a corresponding element in the set of object values B_P. Just as we wish to differentiate between properties of a different type, we also wish to differentiate between their *values* by property type. So in set B_P (the set of object P values) there exists two subsets, S_p (with members that correspond to time-varying values) and C_p (with members that correspond to constant values). The value of the 'number of handles' property $a_{P,0}^C$, maps to $c_{P,0}$ and the 'colour' property $a_{P,1}^C$ maps to $c_{P,1}$. The value of the 'age' property $a_{P,0}^S$ maps to $s_{P,0}$. Note that we are no longer required to employ a *property type* superscript when we refer to property values. We can do this since for example, the S used to denote the subset of properties $A_{P,j}^S$ and corresponding members $a_{P,j}^S$, is also used to represent the set of time-dependent *values* S_P, with elements $s_{P,j}$. Figure 12.3 displays this clearly in diagrammatic form.

12.2.5.2 Event and Time Superscripts

It is sometimes the practice to refer to the value of a variable at different times through the use of a superscript, for example $s_{P,0}^0$ might refer to the 'age' value at time 0 and $s_{P,0}^1$ at time 1. In our model an object can exhibit value changes that are continuously varying (*continuous causation*) and value changes that are event dependent (*event causation*). Note that not all continuously varying values are *explicitly* time dependent. However, for the purposes of this explanation, we shall concentrate on those that are. For this reason we shall employ two superscripts. The first superscript is used to indicate the number of events that have occurred. The second superscript corresponds to the current time index. Let us say we have a value y that is a function of both time and events. The

term $y^{0,0}$ has superscripts which indicate that zero events have occurred and that the current time index value is zero. After one event has occurred we use the term $y^{1,0}$, after two events $y^{2,0}$, and so on. If no events have occurred, but time has progressed (indicated by a time index changing from 0 to 1), we use $y^{0,1}$. If for example two events have occurred, and a period of time Δt has elapsed, we would use $y^{2,\Delta t}$. Note that the second superscript always refers to time, and the first the number of events that have occurred. In order to avoid ambiguity we shall use both superscripts even when the variable is only event dependent, or only time-dependent. We shall only avoid using these superscripts when referring to constant values. Generally speaking, we shall use a notation that is indicative of a system that employs discrete time, although the model that we propose is equally as valid for use with continuous time, with a minor change of notation.

Returning to our earlier example, let us consider again the 'age' property. We shall use the character $\sigma_{P,j}$ to indicate a function that is time dependent, where; P identifies the object to which we refer and j denotes the index of the property $a^S_{P,j}$, and $s_{P,j}$ refers to the time-dependent value (since $S_{P,j}$ represents the set of all values that are time-dependent).

In section 12.2.3.2 when we first introduced the idea of continuous causation, we used $y := f(x)$ to indicate a continuous causal relationship between the causal value x and the dependent value y, where the change in the dependent value y is determined by the function f. We replace this notion with;

$$s^{0,1}_{P,j} := \sigma_{P,j}(s^{0,0}_{P,j}) \tag{12.14}$$

12.2.5.3 A notation for functors

Previously in section 12.2.3.3 when discussing the ways in which continuous causal relationships may be adapted we briefly introduced the notion of functors where;

$$y := f(x) \quad \text{where } f = g[u] \tag{12.15}$$

In this case g represents a functor. The value returned by $g[u]$ is a function, where $g[u]$ corresponds to the selection of an element in a set of functions indexed by the value u. We borrow our notation from the C and C++ programming

languages (where g would correspond to an array of function pointers and f would correspond to a function pointer variable).

We imagine a set of functors (denoted by Θ). An individual element in this set is denoted by the symbol θ_j, where j represents an index into the set of functors Θ. We also define a set of functor states U, whose individual elements correspond to a functor state values u_j. The current value of a functor state u_j corresponds to the associated functor index. For each entry in the set of functors Θ, there exists a corresponding entry in the set of functor states U. The current functor state value for a given functor defines a function σ;

$$\sigma = \theta_j[u_j] \quad \text{analogous to our original } f = g[u]. \tag{12.16}$$

12.3 Reactive Objects

Here we describe the functional structure of entities that exhibit both *continuous* and *event causation*, starting from simple to increasingly complex systems. We use the terms *simple objects*, *reactive objects* and *adaptive objects* to indicate increasing levels of complexity and adaptive potential. This section is concerned with the first two; *simple objects*, *reactive objects*. We shall deal with adaptive objects in the following section (section 12.4).

12.3.1 Elements of an Reactive Object

Figure 12.5 illustrates the internal structure of two *simple objects*; object P contains processes that support only continuous causation, object Q supports event causation, employing our new nomenclature.

In the case of continuous causation (figure 12.5(a)) we presume the existence of a set of functions Σ, elements of which we denote as σ. We also assume the existence of a set of continuously varying values S whose elements are denoted by the symbol s. External objects that are the source of continuously varying values that influence the simple object are denoted by the set $F_{obj.}$, with corresponding elements denoted by the symbol $f_{obj.}$. It is the existence of functions of the form $\sigma(t, s, f_{obj.})$ (where time t denotes time) that determine the response of the object

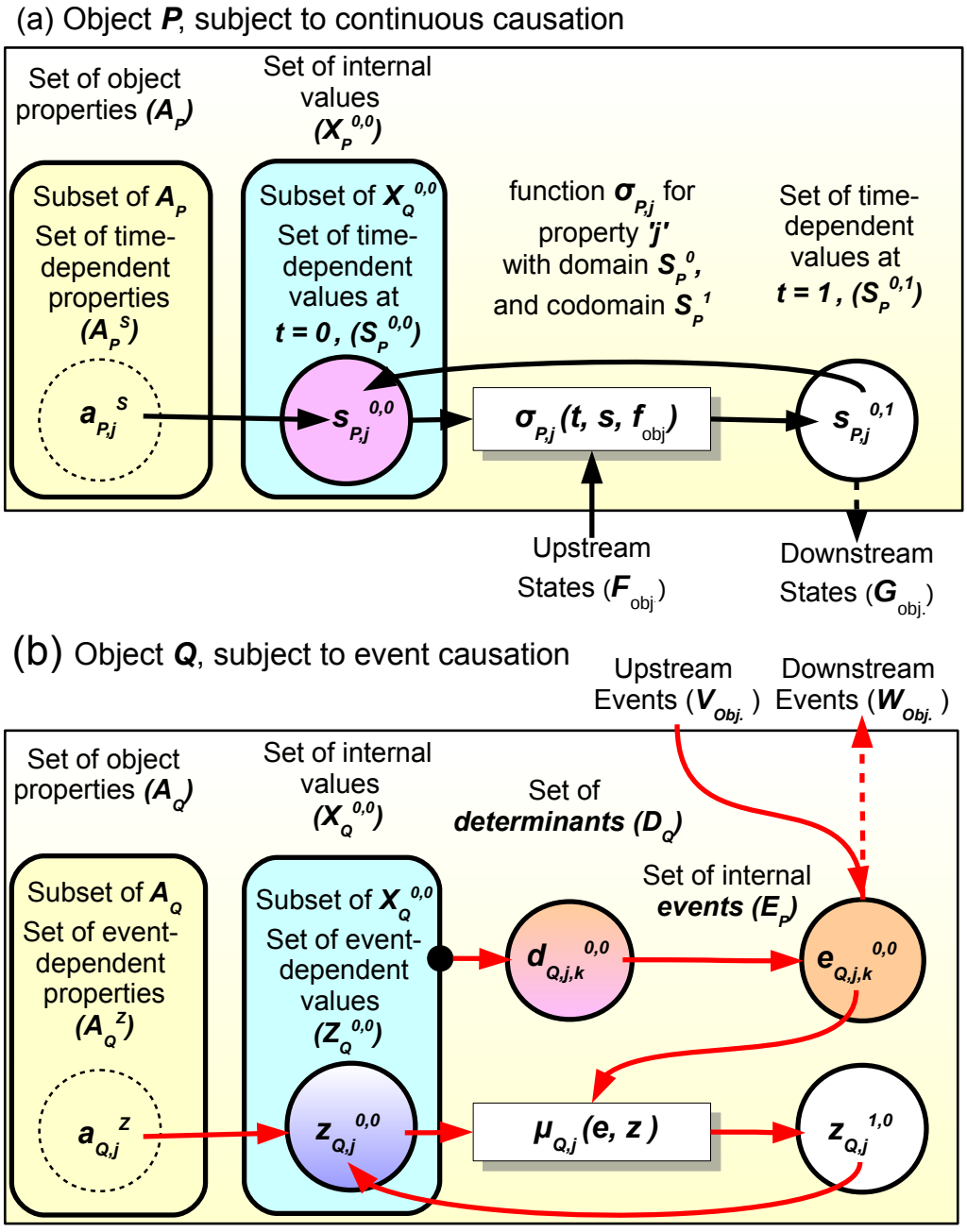


Figure 12.5: Continuous and Event Causation in *Simple Objects*

to continuously varying external values modified by internal state.

The comparable functional structure for simple objects that respond to external events is slightly more complex (figure 12.5(b)). In order to account for varying degrees of causation we presume the existence of determinant functions (δ) that act as feature detectors and event functions (ε) that trigger changes in the object based on the processing of the output of these feature detection functions (see figure 12.2). The determinant functions determine values d (elements of the set of determinant values D) which in turn determine the output of the event functions; the event values e (elements of the set of event values E). Individual state values that are event-dependent are denoted by z, elements of the set Z. We presume the existence of functions μ that determine how the values of Z are updated due to the occurrence of events.

We used to the term *simple* objects in both cases, since they support processes of continuous causation or event causation. What is of greater interest is how two such systems might interact. The detection of the occurrence of an event typically takes place at a discrete point in time, resulting in some step change in some internal state values. In contrast the processes of continuous causation occur continuously. It therefore seems reasonable to presume, that if two such systems were integrated, that it would be the event detection system that is the primary determinant of changes in the continuous causation subsystem.

We discussed earlier in section 12.2.3.3 how processes of continuous causation might be subject to change via parametric or function adaptation. Our concept of a **reactive object** incorporates both of these elements. Figure 12.5 (a) illustrates a hypothetical functional structure where parameter adaptation takes place in response to processes intended to capture aspects of event causation. Figure 12.5 (b) illustrates how processes associated with event causation are able to influence function adaptation in continuous causal processes.

In the case of parameter modification (figure 12.5 (a)), the event-dependent values (elements z of the set Z), are included as parameters in functions that update continuously varying values of the form $\sigma(t, s, z, f_{obj.})$.

Function modification is slightly more complex (figure 12.5 (b)). We imagine that there exist values u (elements of the set of functor states U) that are also event-dependent. We presume the existence of a set Θ, where each element

corresponds to a functor. The functor state arguments determine the current function δ that a functor θ expresses. The result of this is that functions of the form $\sigma(t, s, z, f_{obj.})$, are subject to modification, such that the function itself is a variable whose value is determined by the functor state u; $\sigma^u(t, s, z, f_{obj.})$.

We define a **reactive object** as an entity whose internal functional structure, undergoes both *both* parametric and functional modification in response to the occurrence of events. The functional structure outlined is neither radical nor complex. It represents the simplest possible system that we are able to imagine where processes associated with continuous causation are subject change in response to event causation. The internal functional structure of a reactive object is depicted in figure 12.7, and in a modified version in figure 12.8 (where the internal diagram is intended to depict the functional structure underlying function adaptation).

12.3.2 The Reactive Object from an Ashby Perspective

The whole point of constructing the graph theoretic models of an adaptive object, is to lay out an unambiguous model of how processes of event causation may relate to those of continuous causation, and in doing so present a functional model of primitive brain capable of supporting adaptive behaviour. Whilst an internal diagram of the functional structure of the model has been constructed that delineates this functional structure, there is also some value in comparing our reactive object model with that Ashby's model of ultrastability (referred to in section 2.5.2).

There are a number of congruent features of the two models. The first is the notion of a dual-loop structure. We have recreated the reactive object model in a form that is comparable with that proposed by Ashby (see figure 2.10) in figure 12.10. The structural similarities between the two are immediately apparent. In our model the processes that relate to continuous causation, are directly comparable with those of Asby's 'reacting part'. The processes that relate to event causation replace the functionality that was fulfilled in Ashby's model by the maintenance of 'essential variables' (justification for such a replacement is supported by the work that we recounted in section 5.5.1).

Ashby identified two loops; the first loop, that was responsible for respond-

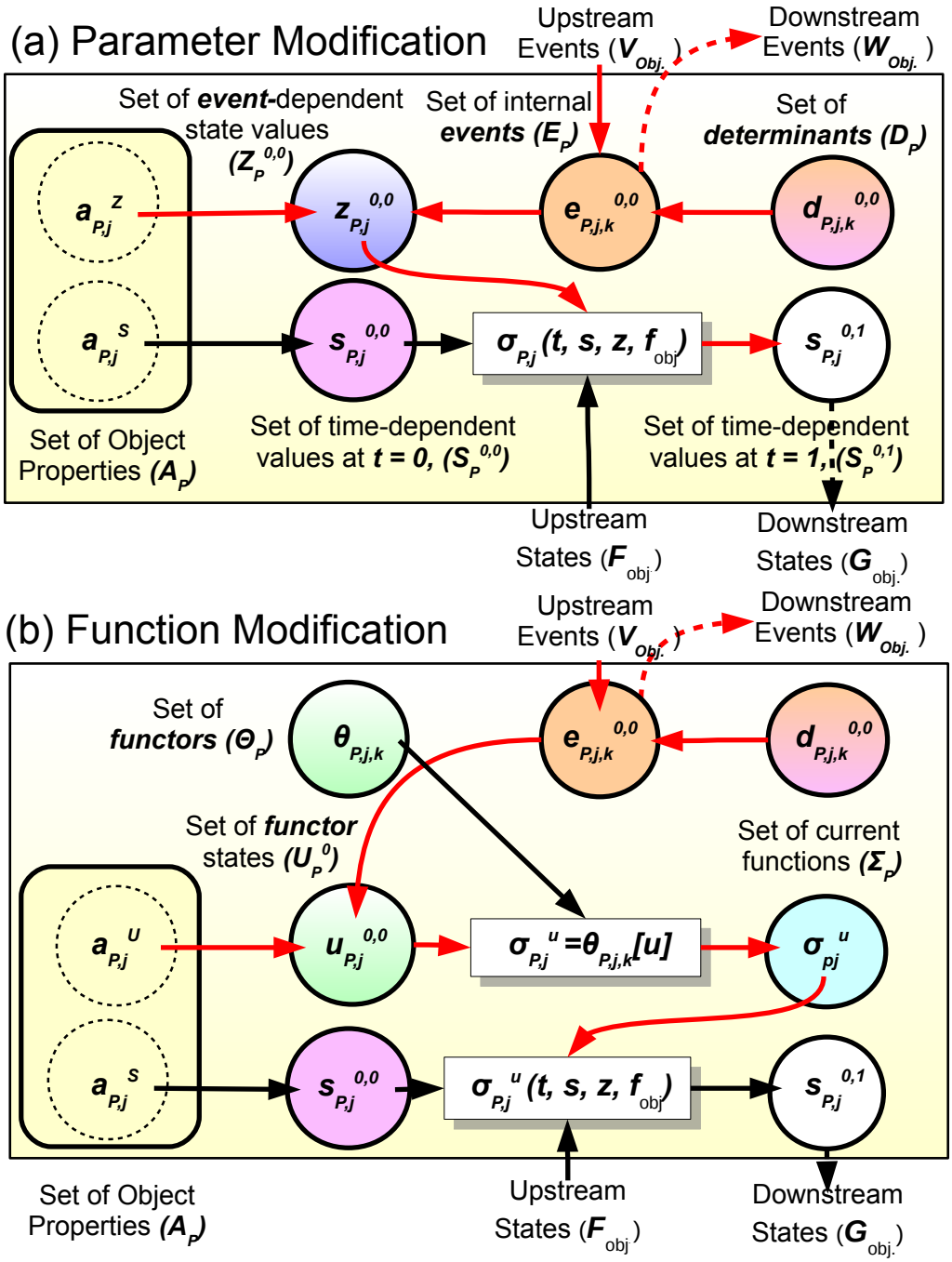


Figure 12.6: Parameter and Function Adaptation in *Adaptive Objects*

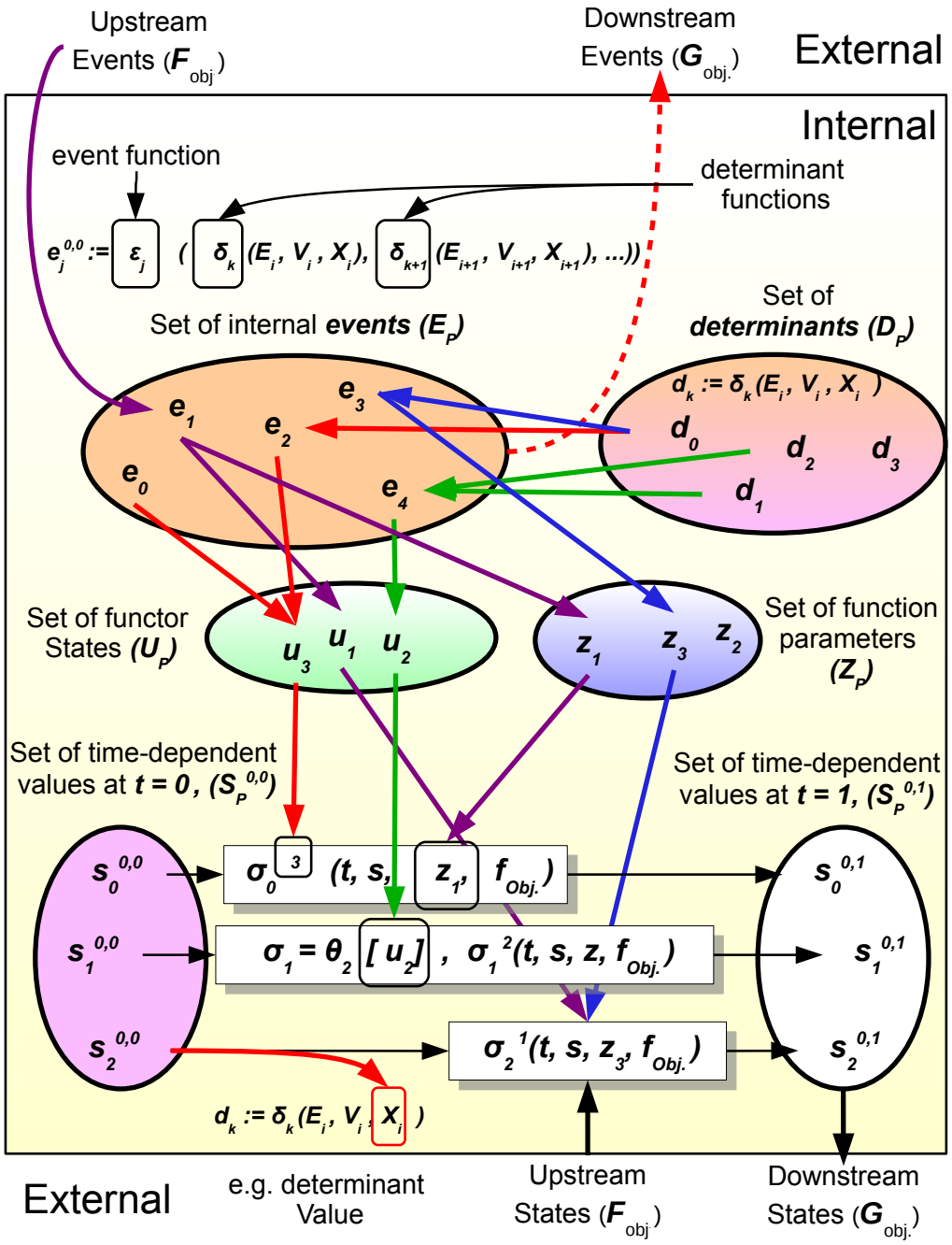


Figure 12.7: **Reactive Object Internal Diagram: Object;** (E) set of internal events,(D) set of determinant values, (U) set of functor states, (Z) set of event-dependent function parameters, (θ) functor instance, (u) current functor state, (S) set of time-dependent values, (σ) current function instance, (z) current function parameters. **Environment;** $(V_{Obj.})$ upstream events, $(W_{Obj.})$ downstream events, $(F_{Obj.})$ upstream time-dependent values, $(G_{Obj.})$ upstream time-dependent values.

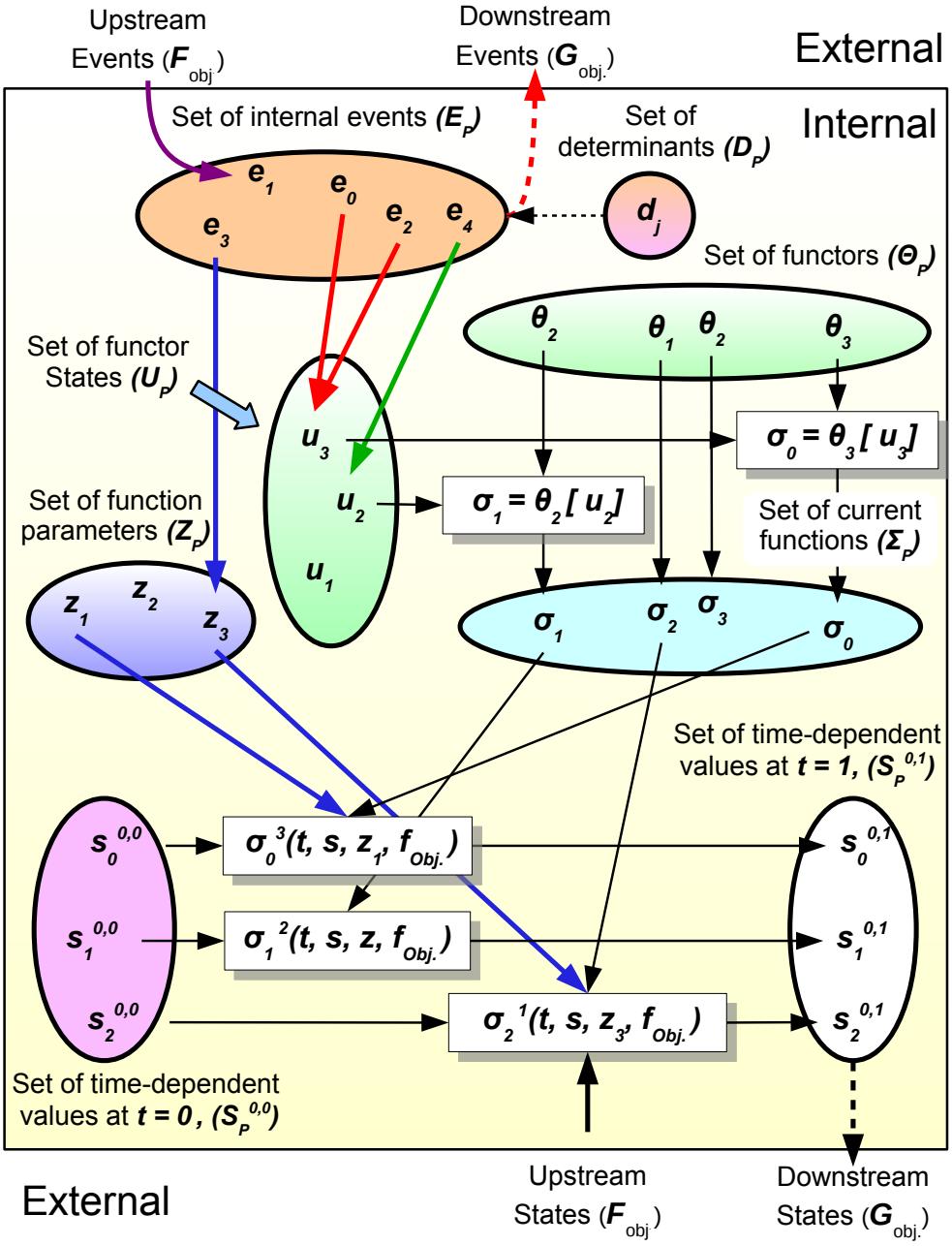


Figure 12.8: **Function Modification Internal Diagram: Object;** (E) set of internal events,(D) set of determinant values, (Z) set of event-dependent function parameters, (U) set of functor states, (u) functor state instance, (Θ) set of functors, (θ) functor instance, (Σ) set of current functions, (σ) current function instance, (S) set of time-dependent values, (s) time-dependent value instance, (z) function parameter instance. **Environment;** $(V_{Obj.})$ upstream events, $(W_{Obj.})$ downstream events, $(F_{Obj.})$ upstream time-dependent values, $(G_{Obj.})$ upstream time-dependent values.

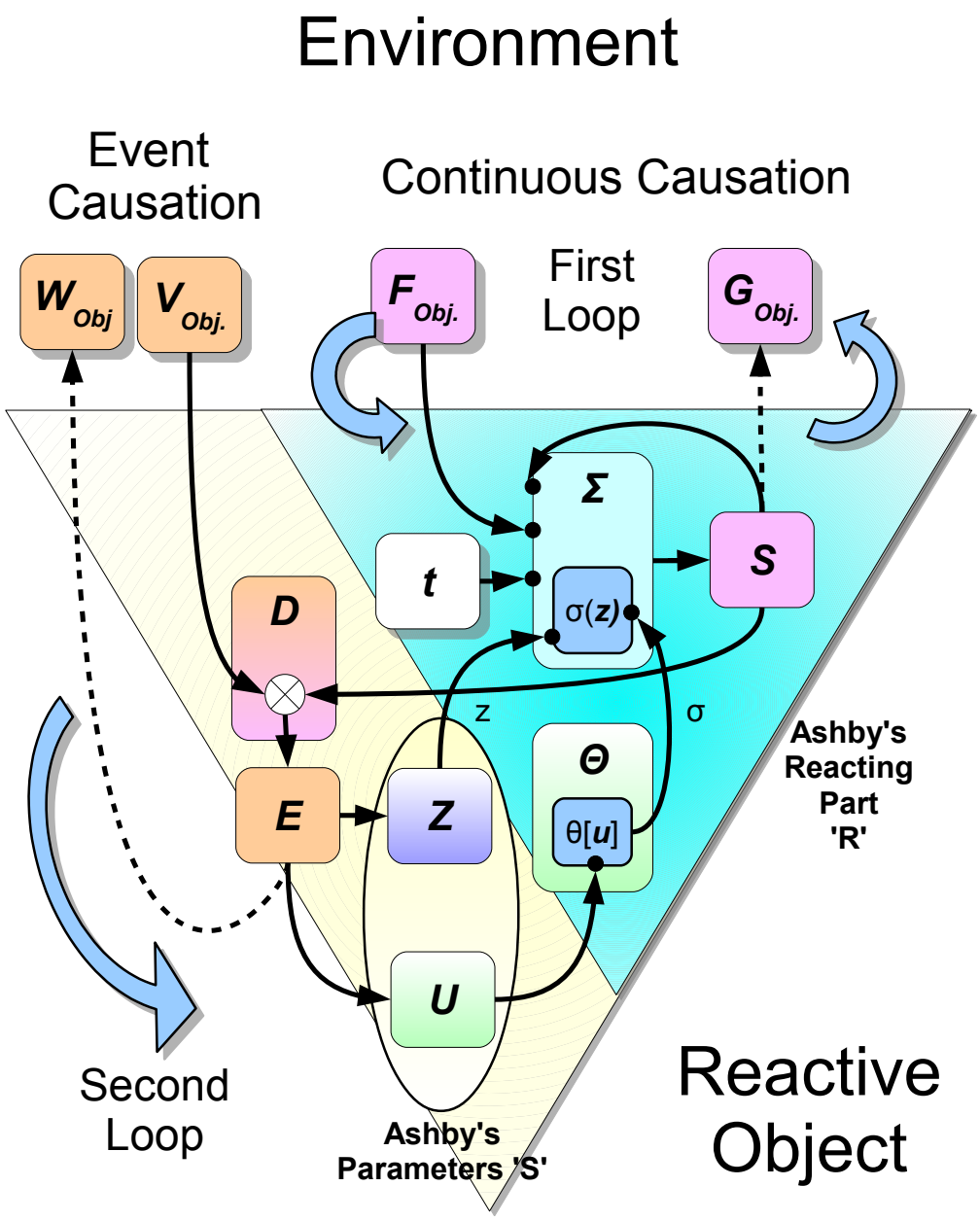


Figure 12.9: **Reactive Object: First loop;** (t) time, (S) set of time-varying values, (Θ) set of functors, (u) current functor state, (Σ) set of current functions, (σ) current function, (z) current function parameters. **Second loop;** (D) set of determinant values, (E) set of internal events, (U) set of functor states, (Z) set of event-dependent parameters. **Environment;** $(V_{Obj.})$ upstream events, $(W_{Obj.})$ downstream events, $(F_{Obj.})$ upstream time-varying values, $(G_{Obj.})$ upstream time-varying values.

ing to changes in the environment, and the second loop, which was responsible for adjusting the parameters of the 'reacting part', when the essential variables transgressed certain bounds. In the case of our reactive object model, the first loop is engaged with processes of continuous causation, and the second loop with those of event causation. Not only are there similarities in the structure of these two loops, there are also similarities in the functional characteristics. Ashby relates the first loop to continuously changing values, and the second loop to step functions or mechanisms. This is entirely consistent with our model the dynamic characteristics of our continuous and event causation sub-systems respectively. The key differences between the two models are the extent to which we define the internal functional structure of the reactive object, and the replacement of the 'maintenance of essential variables' feature of Ashby's second loop. Another difference is in the use of concepts of causation in our explanation of the functional significance of our architecture. We add further complexity to our model when we introduce the notion of the adaptive object in the following section.

12.4 Adaptive Objects

So far in our reactive object model we have delineated a functional structure whereby adaptation of continuous processes occurs through a system of event processing. However, there is nothing intrinsic to this structure that ensures that these changes occur so as to improve that ability of an organism to engage in 'positive' adaptation. In the case of an organism, positive adaptation might refer to the ability of the organism to modify its behaviour such that its chances of survival and reproduction were improved. For this we present the notion of an adaptive object.

An **adaptive object** is one that exhibits the internal functional structure of the **reactive object**, but also incorporates that promote learning (what we have referred to previously as continuous adaptation). This we achieve with the inclusion of two additional functional units; those relating to rewards and memory. The functional integration of these units is displayed in figure 12.9. Their incorporation creates a *third loop*, whereby rewards and memory become part of the causal event processing system. It represents an inner loop, of the

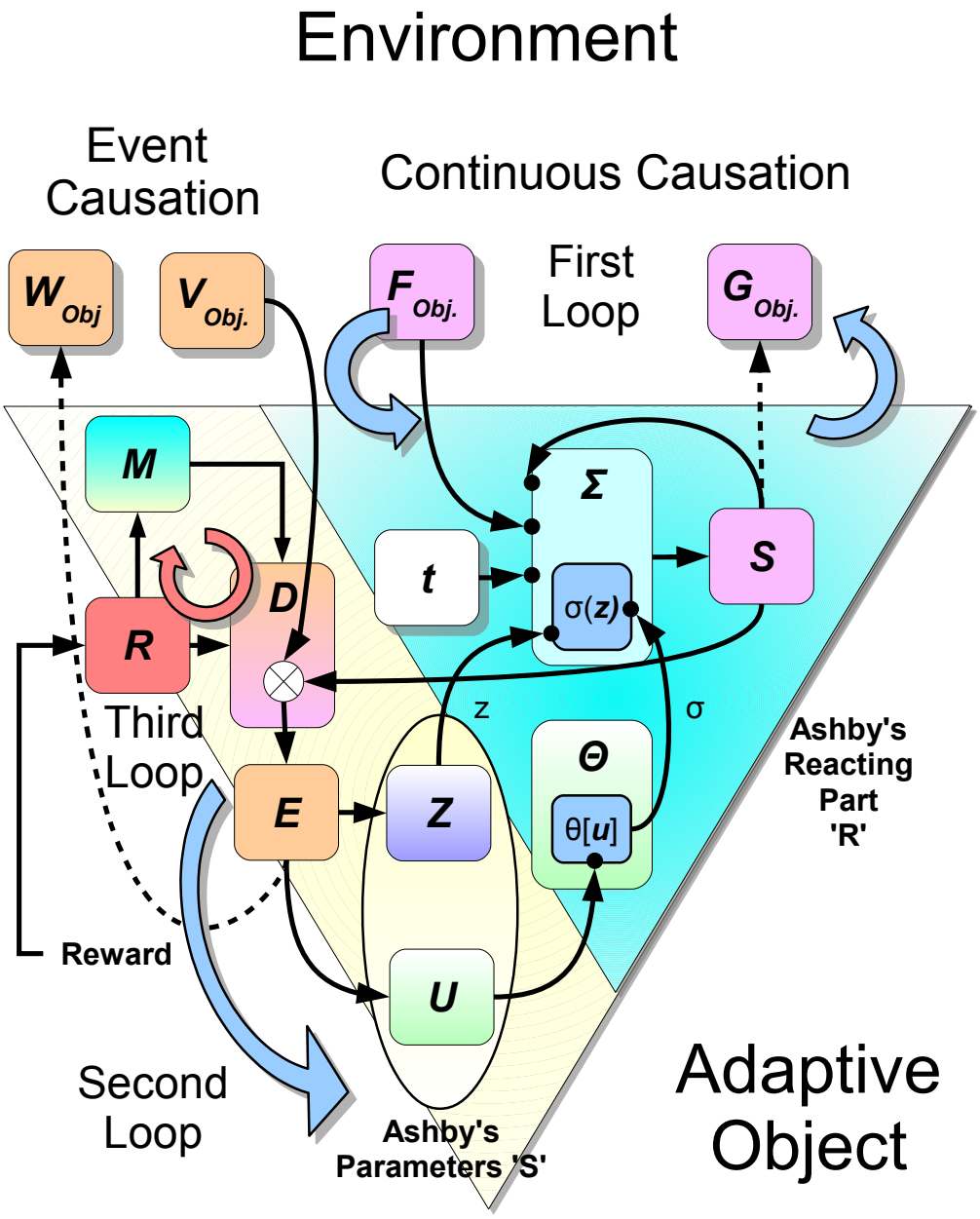


Figure 12.10: **Adaptive Object: First loop;** (t) time, (S) set of time-varying values, (Θ) set of functors, (u) current functor state, (Σ) set of current functions, (σ) current function, (z) current function parameters. **Second loop;** (M) set of value memory, (R) set of reward values, (D) set of determinant values, (E) set of internal events, (U) set of functor states, (Z) set of event-dependent parameters. **Third loop;** (R) set of reward values, (M) reward-related values. **Environment;** $(V_{Obj.})$ upstream events, $(W_{Obj.})$ downstream events, $(F_{Obj.})$ upstream time-varying values, $(G_{Obj.})$ upstream time-varying values.

second loop primarily related to event processing.

Memory supports the persistence of external and internal state values. In doing so, it allows for the possibility of state sequences as an input to the determinant functions (feature detectors), thus allowing for more complex event processing by the event functions. The reward system allows for the possibility of parameter and function modification of the discriminant and event functions, in addition to the role that the system might play in the identification of events.

The incorporating a rewards system into the model provides for some basis for training adaptive objects. This addresses on of the principal weaknesses Ashby's homeostat (see section 2.5.2.2) as an alternative to automatic computation.

12.4.1 The Brain as an Adaptive Object

An obvious question to ask ourselves is whether or not the model that we have proposed bears any correspondence at all with that of biological systems. There is no doubt that such a comparison is highly speculative (and possibly without any merit) but this should not preclude us from at least asking the question. At the very least, it provides us with a model or hypothesis of functional organization in the brain against which other hypothesis can be compared. The results of such a comparison are illustrated in figure 12.11.

- **Cerebellum:** The cerebellum is the part of the brain that is most closely associated with motor control, and as such maps to our notion of continuous causation and the processes that might support it. Afferents to the cerebellum would include in our model; signals that support its parametric and functional modification through event-dependent (z) and functor state values (u) respectively. Note that we briefly described a functional model of the internal workings of the cerebellum, and how this might relate some macro-model of brain functionality in section 5.5.1.

- **Thalamus:** The thalamus is generally characterized as fulfilling the role of a matrix switch or relay. Its nearest correspondence in our model relates to the switching that takes place in the cerebellum due to changes in functor state (u). The determinant of such switching is our event-processing system.

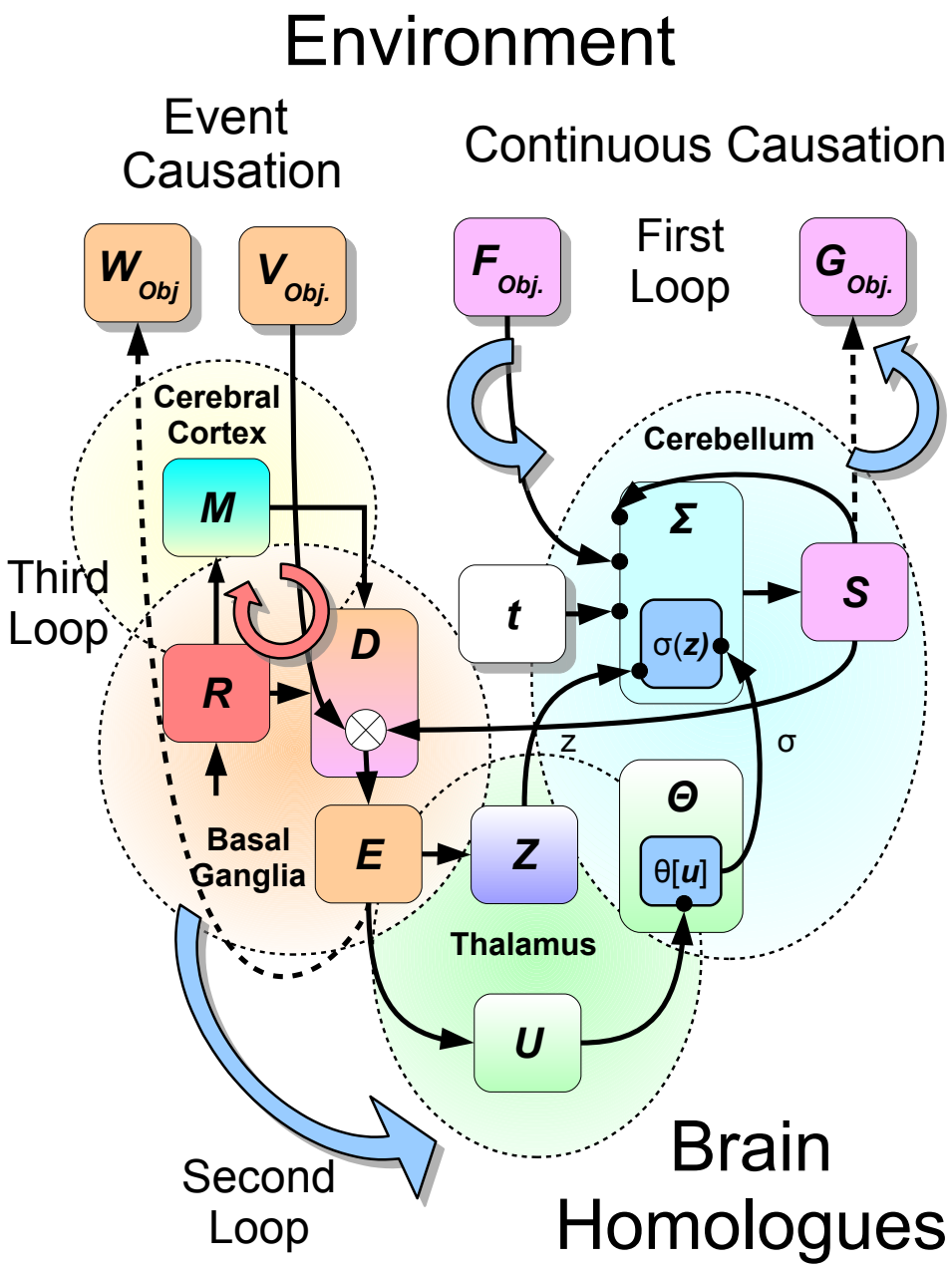


Figure 12.11: **Brain Homologues: First loop;** (t) time, (S) set of time-varying values, (Θ) set of functors, (u) current functor state, (Σ) set of current functions, (σ) current function, (z) current function parameters. **Second loop;** (D) set of determinant values, (E) set of internal events, (M) set of functor states, (Z) set of event-dependent parameters. **Third loop;** (R) set of reward values, (M) reward-related values. **Environment;** $(V_{Obj.})$ upstream events, $(W_{Obj.})$ downstream events, $(F_{Obj.})$ upstream time-varying values, $(G_{Obj.})$ upstream time-varying values.

- **Basal Ganglia:** The basal ganglia is principally associated with reward related activity in the brain. It is most closely analogous in our model to systems that process events, and determine the impact that this has on lower-level (continuous causal) sub-system. Within the basal ganglia system the striatum fulfils the role of our determinant functions, and the globus pallidus fulfils the role of our event functions.

- **Cerebral Cortex:** The cerebral cortex is primarily regarded as supporting associative memory and feature mapping. Both of these attributes correspond to inputs to our determinant functions.

We do not claim that this represents a definitive explanation of the functional organization of the brain at a macro level. Our adaptive object model however, does represent a first-approximation hypothesis. The most obvious criticism that could be made of this model as a hypothesis of brain function is that it does not incorporate the functionality of the hippocampus. The hippocampus is implicated in prediction and sequence-learning (Lisman and Redish [2009] and Okatan [2009], Ginther et al. [2011]), uncertainty (Harrison et al. [2006]), memory (Eichenbaum and Fortin [2009], MacDonald et al. [2011]), navigation (Morgan et al. [2011]), conditioning (Melissa Flesher et al. [2011]), reinforcement-learning (Hirel et al. [2010]) and goal directed behvaiour (Pennartz et al. [2011]). Such a criticism would certainly be true. In fact, this entire thesis essentially ignores the role that the hippocampus might play in reinforcement learning. This is not however intended to signify a lack of importance of hippocampus, only that it is beyond the scope of this thesis. Both the cerebellum and hippocampus are highly complex structures in the brain that no doubt incorporate significant specialized functionality. In contrast with the basal-ganglia the hippocampus (given its functional relationship with the amygdala) is likely to be the key organ that mediates between goals, taking into account both negative consequences and positive rewards. The basal-ganglia in comparison is less complex. An extension of the work described in this thesis would be to seek to develop a more sophisticated functional model of the hippocampus and its relationship with other brain systems.

At least as a first approximation however, the functionality identified by our adaptive object model does appear to correspond (at least in the most vulgar fashion) to that which we know to exist in the brain. The dopamine system in particular is implicated in adaptation in both the cerebral cortex and basal ganglia, representing the primary adaptive signal in our 'third loop'.

12.4.2 Complementary Neural Systems

The idea that the brain comprises complementary neural systems is not new. Grossberg proposed the idea that (originally in Grossberg [2000] and extended considerably in Grossberg [2011]) the brain employs complementary circuits that relate to 'what' (spatial invariant object representations) and 'where' (spatial maps and movement). Whilst this may be the case, the version of complementary functionality to which our adaptive object model corresponds more closely to that that outlined by Doya, where he identifies the cerebral cortx, basal ganglia and cerebellum as distinct areas that are closely related to unsupervised learning, reinforcement learning and supervised learning respectively (Doya [2000]). Together with Capi, Doya propose the idea that the functionality of evolved circuits are adapted to environmental dynamics (Capi and Doya [2005]). This is essentially the idea that we are expounding here, however, we identify causality, in its continuous and discrete forms, as the key world features to which adaptive processes ultimately become shaped.

A common notion relating to complementary functionality in the brain is that of goal-directed and stimulus driven attention (Corbetta and Shulman [2002]). There is evidence to suggest that attention is focussed volitionally by 'top-down' signals, as well as through 'bottom-up' saliency signals (Buschman and Miller [2007], Cabeza et al. [2009] and Cabeza et al. [2012]). This is not inconsistent with our adaptive object model, since the 'top-down' signals correspond to those that originate from the 'third' and 'second loops' (part of our events system), whereas the 'bottom-up' signals correspond to factors that comprise inputs to the determinant functions (that trigger changes in the events system) originating in the systems of continuous causation. Corbetta et al refer to neural systems that support 'reorienting' (Corbetta et al. [2008]), that interrupts and resets ongoing

activity in a fashion that is comparable to functionality supported by our events subsystem.

12.5 Conclusion

The purpose of this chapter is to multiple. In its most general sense it is intended to bring together some of the ideas that have been developed over the course of this thesis, through a general model of an adaptive replicator. In section 2.5.1.1 we identified three adaptive processes (genetic, ontogenetic and continuous) and outlined their integration in what we termed a bauplan for an ***adaptive organism***. An entity that complies with this bauplan and can also be modelled as an ***adaptive object*** we shall term an ***adaptive replicator***. Evolutionary robotics as a methodology is challenging, since it requires a mastery of both these dimensions. We need to conceive of a functional model of the agent's neural systems as a potential target for the evolution of more primitive circuits. The more complex the system, the more difficult this becomes. It is for this purpose, that the adaptive object concept is of particular use. It is an abstraction that describes the macro organization of a hypothetical brain at a level higher than that of the neural circuits themselves.

A superficial glance of this thesis may give the impression that it is disjointed. We started by investigating GasNets and then shifted to reward adaptive circuits. It is however the difference that exists between these two neural network classes that attracts us to them, since our over-riding goal is to determine neural network models that are both complementary and tractable to evolutionary search. GasNets represent a form of neural networks that have characteristics suited to that of the processes of continuous causation. They accept exteroceptive and proprioceptive sensor inputs and generate motor outputs. These are typical of the kind of circuits that we might expect to find in Ashby's reacting part (comprising his 'first loop'). The conditional circuits of the 'second loop' segment and interrupt the sensor-motor patterns of the 'first loop', resulting in significant changes in apparent behaviour. However it is through the 'third-loop' that modulates the discover and replay of these segments. Sensor-motor pattern sequences that result in reward are reinforced so that there is a greater likelihood of being repeated.

Although we know from earlier work that CTRNN's (see chapter 3) are suited to problems of locomotion, other researchers confirm they are less suited to solving problems whose solution represents some analogue of a finite state machine (Phattanasri et al. [2007a]). This mirrors our findings with GasNets reported in chapter 4. In seeking to explain this, the authors state; "Rather, we believe that it was the discrete combinatorial nature of the food edibility learning task that made circuits so difficult to evolve." (p 392, Phattanasri et al. [2007a]). The combinatorial processing of multiple inputs is a core feature of the class of reward adaptive circuits presented in this thesis (as demonstrated in its most simplest form in section 8.4). The combination of these two circuit varieties is therefore of particular interest.

Our adaptive object model helps us to understand how such complementary circuits might be integrated. This integrative role is partly fulfilled by they 'second loop', by the modification of the systems that relate to the processing of continuous inputs, by an event processing system. We identified two routes by which such modification might take place; that of parameter modulation and function replacement. GasNets have the potential to undergo parameter modification through existing neuromodulatory processes. Parameter modification is therefore trivial to implement. In addition, the output of our reward adaptive circuits can be used for the purpose of determining functor state (u values), and therefore function replacement. This could be implemented via a system that supports the selective activation and deactivation of independent GasNet subsystems, in an architectural form similar to that outlined in section 5.5.1).

Whilst by no means complete, the work outlined in this thesis suggests one possible solution to the problem of integrating reward-adaptive circuits with neuron models supportive of locomotion. This may well be sufficient to allow us to make considerable progress in our goal of simulating increasingly complex behaviour.

Chapter 13

Conclusions

This thesis commences with a comparative study of adaptive networks suitable for neuroevolution. GasNets, together with some other forms of dynamic models of neural networks, were shown to be particularly useful for the generation of sensor-motor control circuits suitable for supporting legged locomotion. On discovering that GasNets were less suited to problem domains that require some analogue of a finite-state machine, an attempt was made to hand-code a solution to such a problem, and in so doing, identify architectural features that might be significant in solving such a class of problems. In comparing these features with biological systems, the basal ganglia appeared to be a structure of particular significance, and by implication, the role of dopamine as a neuromodulator implicated in reinforcement learning.

One particular aspect of dopamine signalling appeared to offer a clue as to the underlying processes that might regulate dopamine phasic signalling. This is the effect that is observed, when dopamine signalling levels drop significantly lower than basal levels, when an anticipated reward fails to materialize. Speculation that this might be explained by some process whose function might be to support energy efficient signalling, led us to the creation of a novel neuron model.

A small conceptual leap enabled us to imagine how a small group of such neurons, in competition, could support a tri-modal action selection system, incorporating; **selection**, **exploration** and **exclusion** (see chapter 8). Similarities with the role of the medium spiny neurons (MSN) in the striatum are striking, potentially providing us with a functional explanation for the *up-down* states that

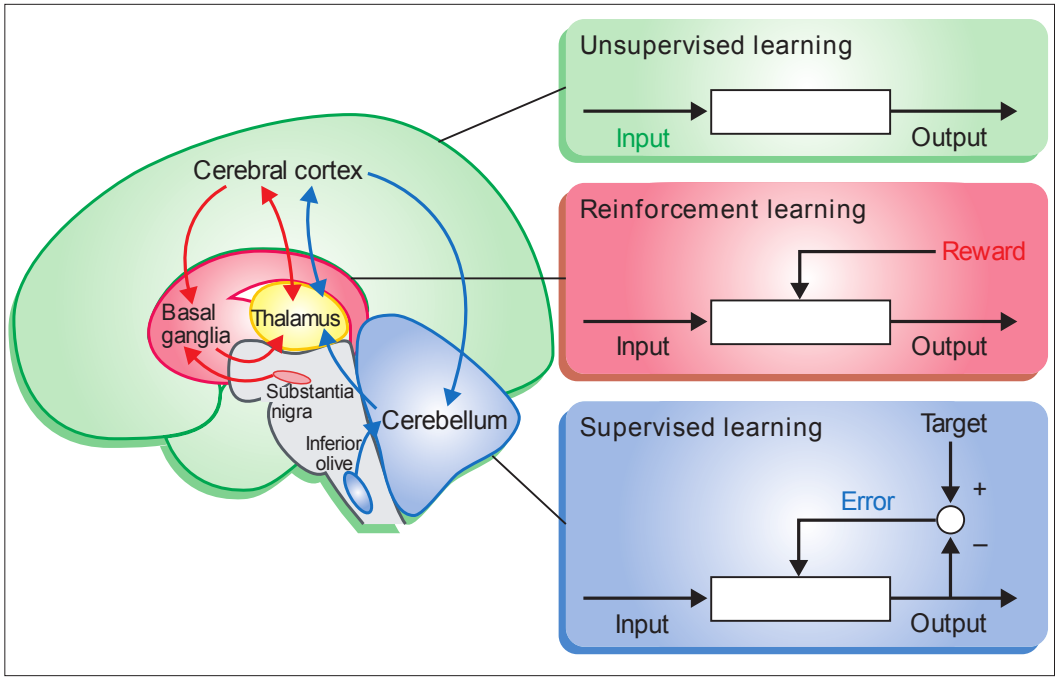


Figure 13.1: **Categories of Learning in the Brain:** This schematic was presented by Doya to illustrate the complementary roles of basal ganglia and cerebellum. **Supervised learning** is commonly implemented through multi-layer perceptrons. **Unsupervised learning** is implemented by self-organizing feature maps. The circuits presented in this thesis comprise a new class of **reward adaptive** circuits that fulfil the role of **reinforcement Learning**. Source Doya [2000].

are characteristic of these neurons. It is interesting to note that these patterns of activation are also prevalent in the cerebral cortex (Holcman and Tsodyks [2006]), potentially implying some functional correspondence between the two. There is evidence to suggest that this is also modulated by dopaminergic projections (Lewis and O'Donnell [2000] and O'Donnell [2003]).

Circuits comprising these model neurons (which we sometimes refer to as Rayleigh neurons), have as an intrinsic feature, a reduction in excitation of the source neuron with a concomitant increase in responsiveness of the target neuron), as part of an energy regulatory regime. This occurs when an important input-output (sensory-motor) association has been made. The concept of valued or significant association pathways undergoing inhibition provides us with a potential explanation for the prevalence of inhibitory pathways in the striato-thalamic loop. This model of excitation reduction in our neuron model (for the purposes of supporting energy efficient signalling), potentially changes our perspective on the role of non-local inhibitory pathways from one of *blocking* or *stopping* to one of dynamic channel regulation through amplitude modulation.

In a more general sense these results are supportive of the idea that *intrinsic* rather than *task-evoked* signalling should be given greater consideration (Raichle [2006]) in seeking to understand the functional nature of the brain. The proposed neuron model supports a hypothesis that under certain circumstances neural signalling may take place close to some basal limit, and that excitation levels and synaptic weights are dynamically adjusted to maintain energy efficient signalling (a simple example of which is demonstrated in section 8.4). Whilst chapter 4 represents a crude attempt to incorporate energy efficiency into fitness measures (from the perspective of energy expended in action), a shift in focus to the energy efficiency of brain signalling systems may provide us with a more appropriate basis upon which to evolve artificial neural circuits that are closer to those found in biological systems.

Figure 13.1 illustrates three key brain regions and the categories of learning that we associate with them (Doya [2000]). We associate the cortex with *unsupervised learning*. Self-organizing feature maps are often used to model such systems. We associate the cerebellum with *supervised learning*. We often model such systems with multi-layer perceptrons. Whilst the basal ganglia is associ-

ated with *reinforcement learning*, we do not have a class of simple connectionist models that capture the characteristics of this system. Reinforcement learning problems are usually solved through techniques and algorithms that originate from dynamic programming or control theory. This thesis proposes a number of simple circuits that with further development could provide us with connectionist models that are useful in supporting reward adaptive learning.

Evolutionary robotics is at its foundation biologically inspired. It seeks to prioritize biologically plausible models over more abstract models. Our ability to extend this field of work is reliant upon our understanding of biological systems (from which inspiration is drawn) and models that adopt the appropriate level of abstraction. The models and circuits proposed here, are neither overly complex, nor do they lack biological plausibility. At the same time, they complement functionality that is already adequately provided by GasNets and other adaptive networks for the purposes of robotic simulation. Work that combines these two approaches offers us the potential of supporting increasingly complex behaviours.

We conclude the thesis with the portrayal of a formal model that may be of use in understanding how complementary circuits might be integrated. This model is inspired by the work of W. Ross Ashby (referred to principally in sections 2.5, section 5.5.1 and in chapter 12), whose ideas help to provide a common perspective from which the seemingly disparate areas addressed in this thesis can be seen to be part of a coherent whole.

Appendix A

Software Application Framework

A.1 Application Framework Overview

The vast majority of time expended on the creation of software designed to support the work described in this thesis went into the creation of an application framework in C++. It was designed from the start to be a framework targeted to support high-performance simulation applications. It is a modern architecture that has been designed with multi-threaded programming in mind. Through a high degree of isolation between components it is also targeted at development teams who may wish to work on individual components without the need to re-compile the entire application. This supports large-scale development teams who may well be geographically remote from one another.

This framework is intended to achieve the following goals;

- **Flexibility:** Applications can be composed in their entirety from a simple application meta-program script. A minimal run-time engine instantiates the script into a full application. Cutting and pasting of the segments of the script allows for the creation of entirely new applications with a maximum of code reuse.

- **Performance:** The use of computational resources is optimized through

the use of multi-threading techniques in both component creation and communication.

- **Simplicity:** The complexity of multi-threaded programming is hidden from the user by the extensive use of threaded base classes.
- **Efficiency:** Serialization classes and utility tools are provided for the automation of labour intensive programming activities.
- **Spreadability:** Through the use of binary buffers and the support for asynchronous and synchronous communication it should be relatively trivial to convert an application from a stand-alone to a distributed application.

The main features of the framework are summarized below;

- **Plug-in Framework:** Plug-in based architectures are common in high-end commercial applications where program modules are loaded dynamically at run-time. Users are allowed to create their own plug-ins that significantly extend and expend component functionality.
- **Meta-program Language:** A meta-program language was created (based in Lua) that specifies the components that create an application and the dependencies between them. This gives us a high-level abstraction of program modules and allows for the maximum flexibility in the creation of applications through the composition of plug-in modules. A meta-program component exists that can instantiate its own application, supporting the hierarchic composition of meta-programs.
- **Binary Serialization:** These classes allow for the automatic serialization of arbitrary data. These classes are used extensively in the framework, supporting; the automatic display and editing of data in user interface components, a common standard for communication between components.
- **Component Communication:** Communication between components can either be synchronous or asynchronous. All communication between components takes place via memory-mapped files which act as buffers for use

by the binary serialization classes. Applications can be composed of components that are coupled either synchronously or asynchronously.

- **Threaded Components:** Threaded component classes comprise the building blocks of the application framework. The application framework has been designed and written from the start to support mult-threaded applications.

- **OpenGL Components:** Real-time visualization classes have been written in OpenGL. Note that under windows OpenGL windows each have their own thread, allowing for the construction of highly efficient visualization applications on modern processors (each OpenGL render window has its own thread).

- **GUI Components:** GUI components have been written primarily to support spread-sheet like parameter editing and simulation controls. They have been written to allow for the integration of the application framework *within* and existing windows based application (allowing for the creation of child windows), as well as allowing for the creation of top-level windows.

- **Physics Simulations:** Components exits to support 3D (NVidia's PhysX library) and 2D (the Box2D Library) simulations.

- **Utilities** A code generator based on text templates has been created for use in the creation of custom buffer classes (with automatic serialization built-in).

A.2 The Meta-program Language

The application framework unique in that an entire application can be instantiation from a simple text based script. This script comprises a meta-program language that acts as a high-level description of the application in terms of a number of *components* and the dependency relationships between them. A component is a class instance created from a class factory contained in a Dynamic Link Library

(DLL). On first running the run-time framework first examines the local directories for a *plug-in manifesto* (a text list of accessible DLLs housing components). If one is not found, then one is created by first searching the local directories for compatible DLLs. The plug-in manifesto stores the name and location of each component accessible by the application. On parsing the meta-program script the relevant DLL's are loaded into application memory, components created, and the dependencies between them instantiated. This section gives an over-view of the meta-program script and how it relates to the composition of an application.

The meta-program script is written in Lua (an embeddable scripting language). The meta-program script is written with a syntactic structure that acts as a declarative language for the composition and structure of a component based application. However, we are able to make use of the Lua run-time interpreter for the execution of commands that allow for the dynamic modification of the application. This description however focuses on the key characteristics of the meta-program script in its declarative form. For readers unfamiliar with Lua syntax we have also included C++ like pseudo-code listings to help explain the significance of the meta-program code blocks.

A.2.1 Structure of a Meta-program File

A meta-program file comprises three key elements (see A.1);

1. **Alias Table(s):** The first part of a meta-program is a list of *aliases* to which we can assign a component **type**. Quite simply, an alias corresponds to a component type variable. Each component has a single string (written as a literal string, corresponding to the components *name*) which we can interpret as the *type* of the component. If we wish to update the application to use a modified component, all we have to do is to assign the new component name to the alias of the component type that we wish to replace. This allows us a single point of change when modifying a meta-program script to replace an existing component. It also provides us with a single reference location for examining what components this application relies on. We refer to this meta-program section as the *table of aliases* or *alias*

table. It is possible to have multiple alias tables in a single meta-program script.

2. **Instance Table(s):** This is the heart of the meta-program. A single *instance table*, contains a number of *collections*. In each collection we have (1) the names of instances of components to be created (paired with its *type alias*), (2) the dependencies that exist between these components. We refer to the declaration of instances as a *instance list*, and the declaration of dependencies as the *dependency list*. A single instance table may have within it multiple *collections*. We are able to differentiate between *local* (component instances in the same collection) and *non-local* (component instances in other collections) components in a *dependency list* through minor changes in syntax. In the table of aliases we have an alias for every component type. In a single collection we are able to create multiple instances of the same component type, and differentiate between these instances when we specify the component dependencies.

3. **Application Table:** Whilst a meta-program script may contain a number of *alias tables* and **instance tables**, it will only ever contain one **application table**. The application table lists the *alias* and *instance tables* that comprise the application. When the meta-program script is parsed, this is the first table that is inspected. The names of the alias tables included here will first be parsed, before the named instance tables are parsed and components created. The final stage of parsing is to instantiate the dependencies that exist between created components.

The meta-program language is intended as a high-level language that hides much of the implementation details of the components, whilst maximizing the flexibility of application creation through component composition. A key part of this process is in the high-level description of dependencies between components. The meta-program describes the dependencies that exist between components in the *dependency lists* present in each *collection* (found in the *instance tables*). Currently there are four major dependency types;

1. **Windows:** This dependency type relates to graphical user interface (GUI)

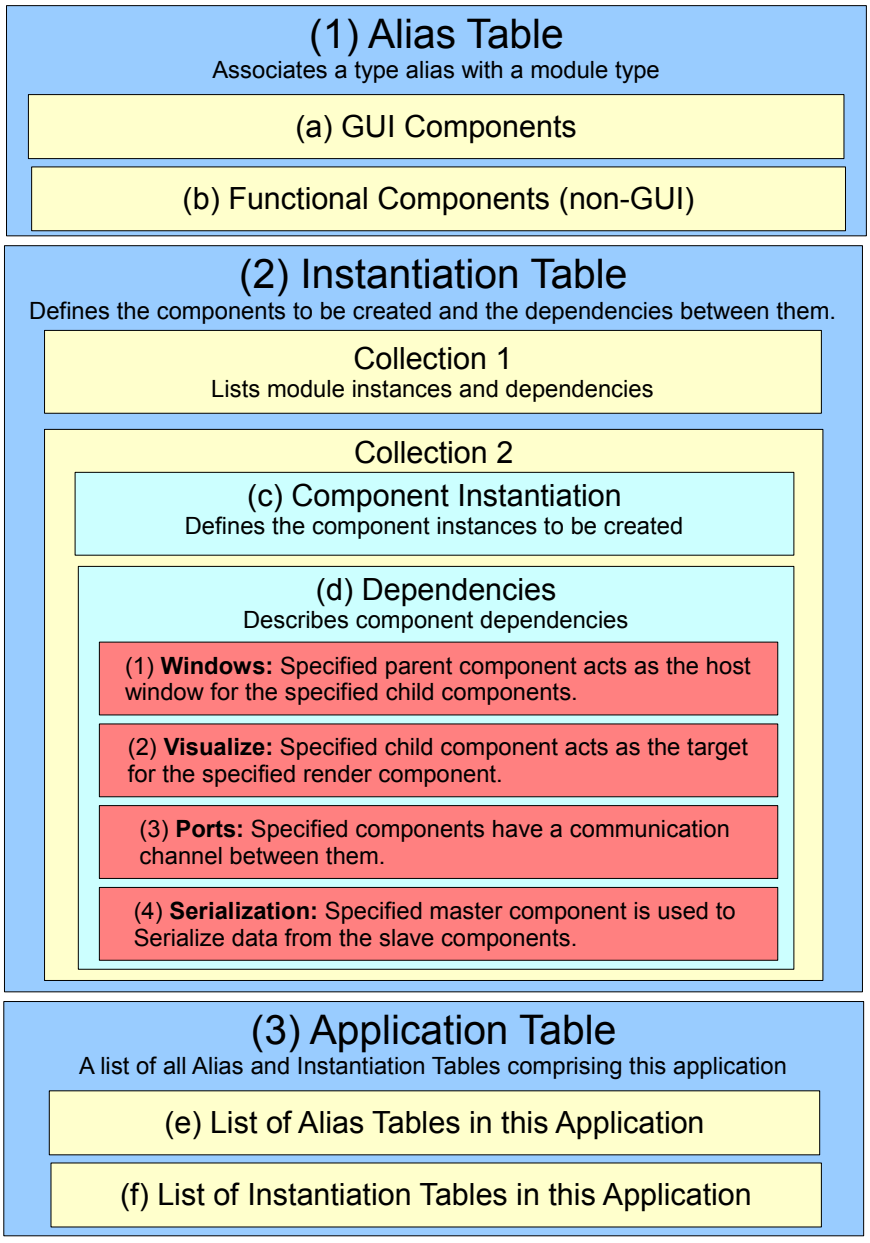


Figure A.1: Meta-program Structure

components. Currently there exist two component types that create windows that act as a host for other meta-program components which create child windows. One of these components creates a top-level window that acts as a host. The other creates a host window *inside* an existing Windows application. In either case, we are able to specify a dependency relationship between the *parent* host window and any child windows created by other GUI components. The role of these host windows is simply to manage and arrange the child windows attached to them by this dependency relationship. All windows management is taken care of by these host windows and existing child window components.

2. **Visualization:** There is one particular GUI component type that requires an explicit dependency associated with it. This is where we have an OpenGL render component. The render component does not of itself create a child window. It requires a separate child window component to act as a render target. This child window also reflects back any user interaction to the render component to allow it to modify the visualization display. One of the reasons for the separation of the functionality of the child window and the OpenGL render component is to allow for a more flexible application framework. A standard child component allows us to route user interactions to a render component without knowing how the image is to be prepared. It allows us to replace the render component without modifying the way in which the user interacts with the display.

3. **Ports:** At the core of meta-program application framework is the notion of *ports*. A port is a bidirectional communication channel created between specified components, allowing for synchronous and asychronous communications between components that may or may not be executing independent threads. Port implementation details are described in section A.3. Communication through ports takes place through standardized binary serialization buffers. The standardization of these buffers allows us to support communication between components without needing to know the implementation details of either the transmitter or receiver. Providing the binary layout of the serialization buffer remains constant, we are able to replace individual

components significant upgrades without the need to replace the alternate component that constitutes the communicating pair.

4. **Serialization:** Since the meta-program script supports the arbitrary composition of components, it is useful to have a standardized means by which program state data can be saved to and retrieved from files. We have created specialized components who act as managers for the serialization of application wide (and component specific) data. We specify the name of the component instance that is to act as the serialization manager and serialization support will automatically be available to meta-program instances should the need use of it.

A.2.2 A Minimal Simulation Application

As an aid to understanding we shall describe the key meta-program features in the context of a minimal application comprising five components (see figure A.2). This program comprises many of the key components that we might expect to find in an application intended for simulation, and is based on a real application used to investigate the characteristics of Self-Organizing Feature Maps (SOFMs).

The five components include;

1. **Master Window Component:** that acts as a top-level window under the Windows operating system. Note that although we use a top-level window here, it is possible to embed the entire framework within an existing application providing we have access to a single window as host.

2. **Simulation Component:** This component hosts the simulation model for the SOFM. It is threaded component that owns its own control and worker thread. It is this worker thread that is responsible for carrying out the simulation of the SOFM model.

3. **Grid Component:** On start-up the simulation component first publishes the model parameters to the grid component. The user is then able to modify these parameters by modifying the entries in the grid component.

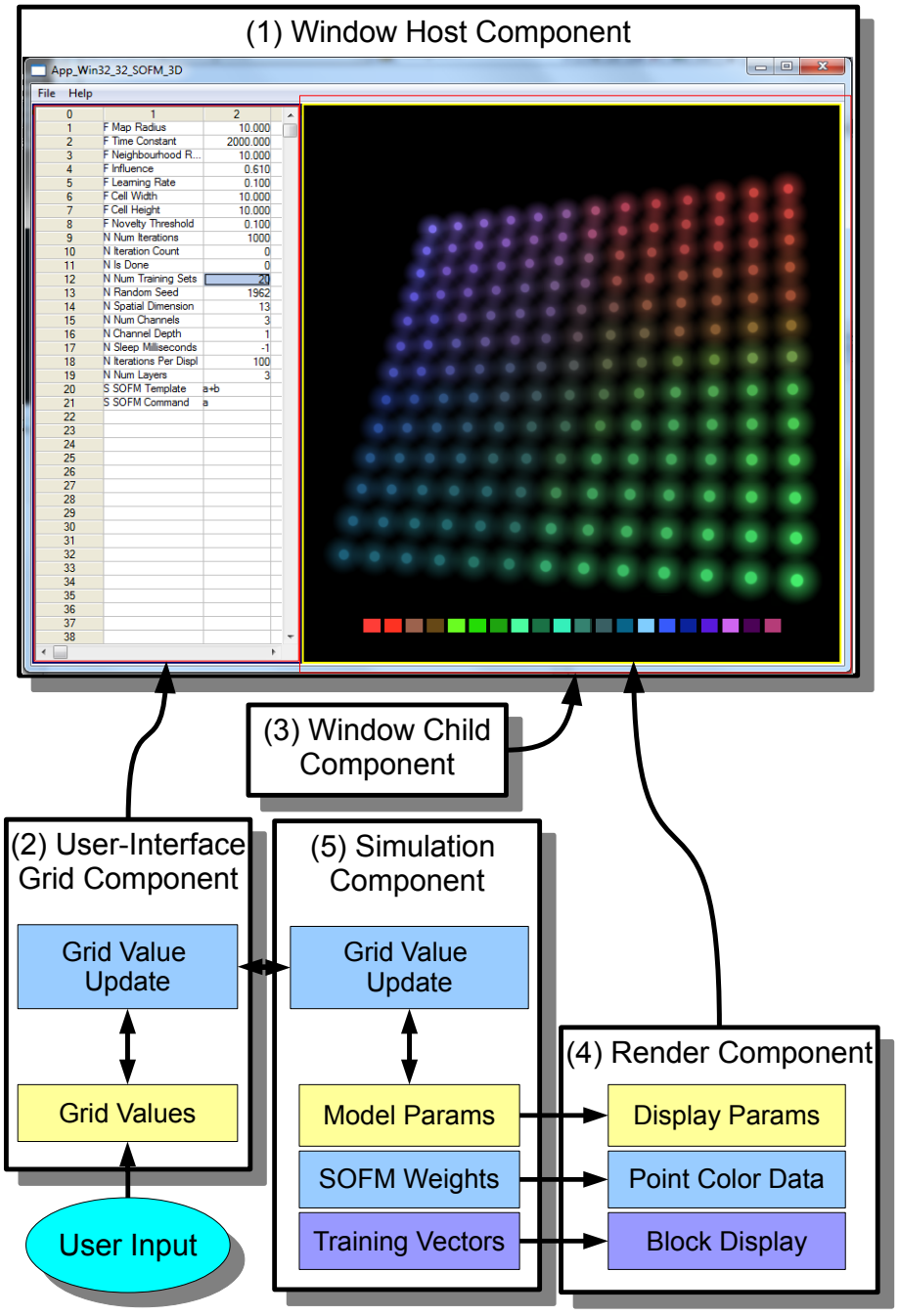


Figure A.2: Components for a minimal simulation application

These modifications will be automatically reflected back to the simulation component.

4. **Visualization Component:** For the purposes of understanding how the model is performing we use an OpenGL render component. This component allows us to render the SOFM in a 3-dimensional space. It also displays a strip corresponding to the input vectors used to train the SOFM. The OpenGL render component has its own thread.

5. **Child Window Component:** The visualization component is unable to display anything without having a child window as its render target. This child window allows for a seamless integration with the Host window (in this case created by the Master Window Component). User interaction with the child window is fed back to the visualization component so that the user is able to interact with the display of the SOFM (in this application the SOFM can be rotated in 3 dimensions).

Writing such an application from scratch would require an extensive knowledge of both the Windows operating system and multi-threaded programming. There are 4 threads intrinsic to this application; the main process thread (responsible for maintaining the user-interface and process Windows messages), a control and worker thread in the simulation component, a render thread in the visualization component. However, all these components are available to the programmer for modification and customization, with windows and multi-threading details taken care of by existing classes or the meta-program run-time application. Updating the application is a simple as replacing the functionality comprised on one or more of the existing components.

Pseudocode for the meta-program script corresponding to this application is shown in listing A.1. The key elements comprise the

- Alias Table: Here we can see the names of the component types that are to be created.

- Instance Table: The *instance table* lists the instances of components to be created, and specifies certain dependency relationships between them. Here

we can see the creation of a Parent-Child window dependency, the association of an OpenGL render component with a render target (a child window component), the creation of three communication ports between the simulation and render component, and the creation of a single communication channel between the parameter grid GUI component and the simulation component.

- Application Table: The application table stores the list of alias and instance tables comprised in this application.

Listing A.1: Meta-Program Pseudo Code

```
Aliases Types_01
{
// (a) GUI Components
<Host_Wnd> = "GM_Host_Wnd"
<Chld_Wnd1> = "GM_Args_Grid"
<Chld_Wnd2> = "GM_Rend_Child"
<Sofm_Rend> = "GM_Rend_OpnGL"
// (b) Functional Components
<Sofm_Rend> = "GM_Rend_OpnGL"
}

Instances Instances_01
{
        Collection Collection_01
        {
        // (c) Instances
        <Host_Wnd> Master_Wnd;
        <Chld_Wnd1> Common_Grid;
        <Chld_Wnd2> Rend_Chld;
        <Sofm_Rend> SOFM_Rend;
        <Sofm_Rend> SIML_SOFM;

        // (d) Dependencies
        Dependency("Parent_Chld_Window", Master_Wnd, Common_Grid, Rend_Chld );
        Dependency("Parent_Chld_Render", Rend_Chld, SOFM_Rend);
        Dependency("Port_Attach_Down_Stream", SIML_SOFM, SOFM_Rend, 3);
        Dependency("Port_Attach_Down_Stream", Common_Grid, SIML_SOFM);
        }
}

Application App_01
{
List_Aliases(Types_01);
List_Instances(Instances_01);
}
```

A.2.3 Alias Table

The Alias table is where we define the component types. Listing A.2 shows how this would appear in C++ like pseudocode, and listing A.3 shows the actual implementation in Lua code. The function of this table is very simple. We are simple assigning component types to aliases that we will use in our Instance table. This allows us to rapidly change the component type without modifying any other part of the meta-program script. This is intended to support the use case in many research applications where we wish to incrementally modify an existing algorithm, update the display of the algorithm results, or try alternative formulations. Typically the dependencies between other components remains the same.

Listing A.2: Alias Table in Pseudocode

```
Aliases Types_01
{
// (a) GUI Components
<Host_Wnd> = "GM_Host_Wnd"
<Chld_Wnd1> = "GM_Args_Grid"
<Chld_Wnd2> = "GM_Rend_Child"
<Sofm_Rend> = "GM_Rend_OpnGL"
// (b) Functional Components
<Sofm_Rend> = "GM_Rend_OpnGL"
}
```

In C++ code there exists the concepts of templates. In an analogous fashion the type variable used in template classes acts as a place-holder for a concrete type. Here the alias acts a place-holder for a component type. The corresponding Lua code is more verbose. In the Lua code we are also able to define default parameters for GUI components. This is particularly important for the Window

Host component as this component specifies the an arrangement of child windows via a layout template string. The Lua script also allows us to specify a file name as an argument. The string associated with this file name will be passed to the component as part of the component initialization process (thus allowing for component specific files to be used for initialization purposes). We are able to assign different alias names to components of the same type. This allows us to create initialization parameters for individual components of the same type.

Listing A.3: Alias Table in Lua

```
App_DEFN=
{
["_TABL_Name"]="Types_01",
["_COMP_Info"]={
        ["_Script_Name"]="Default_Script",
        },
["_COMP_GUI"] ={ -- GUI Components
  ["Host_Wnd"]={
                ["_Comp_Name"]="GM_Host_Test_Wnd",
                ["_Comp_Args"]={
                        ["m_wTitle"]= "Host_Window", -- Window Title
                        ["m_nX_Off"]= 0, -- Left Offset
                        ["m_nY_Off"]= 0, -- Top Offset
                        ["m_nWidth"]= 970, -- Width
                        ["m_nHeigt"]= 546, -- Height
                        ["_Layouts"]= {
                                ["_Layouts_List"]= {
                                        {
                                        ["_Layout_Name"]="Default",
                                        ["_Layout_Tmpl"]="a+2.0b",
                                        },
                                },
                        },
                },
        },
```

```
["Chld_Wnd_Lft"]={["_Comp_Name"]="GM_Grid_Multi_Table",},
["Chld_Wnd_Rgt"]={["_Comp_Name"]="GM_Rend_Child",},
["OGL_2D_Parts"]={["_Comp_Name"]="GM_EGn1_SOFM_Cam",},
},

["_COMP_FNC"]={-- Non-GUI Components
  ["SIML_SOFM"]={["_Comp_Name"]= "GM_EGn1_World_Sim",},
  },
}
```

A.2.4 Instance Table

The instance table is the heart of a meta-program script where we specify the components to be instantiated together with the dependencies that relate to them. In the pseudocode listing of A.4 this is very succint. We simply associate an instance name with a type alias. Subsequently we refer to the instance name when we specify the dependencies between components. This is essentially the same semantic construction as that of the Lua code listed in A.5. The only significant addition is the inclusion of a initialization details for the master window component. Here we refer back to the window arrangement specified as a template string in the Alias Table entry for the host window. In the instance table we assign specific component instances to the tokens specified in the Alias Table. This ensures that the GUI components are assigned the appropriate dimensions and relative positions when they are created. Note that the syntax used for the alias types corresponds to that used in Lua when we are referring to entries in other tables. The dot notation allows us to make a global reference to an external table (in this case the Alias Table).

Theoretically we could have multiple alias tables referenced by the same in-

stance table. The dependency section of the meta-program lists four dependencies. The *Parent-Child Window* dependency refers to the relationship between a host window and the child windows of the GUI components. The *Parent-Child Render* dependency associates an OpenGL render component with a specific child window. This child window becomes the render target of the OpenGL component, in turn the child window routes user interaction (either mouse or keyboard commands) from the child window to the render component for display modification. The *Port Attach Down-Stream* dependency type is used to create communication channels between the simulation component and firstly the visualization (render component) and secondly the grid component (used to display simulation model arguments, and support parameter modification). Further details of these communication ports are display in figure A.4.

Listing A.4: Instance Collections in Pseudocode

```
Instances Instances_01
{
        Collection Collection_01
        {// (c) Instances
        <Host_Wnd> Master_Wnd;
        <Chld_Wnd1> Common_Grid;
        <Chld_Wnd2> Rend_Chld;
        <Sofm_Rend> SOFM_Rend;
        <Sofm_Rend> SIML_SOFM;
        // (d) Dependencies
        Dependency("Parent_Chld_Window", Master_Wnd, Common_Grid, Rend_Chld );
        Dependency("Parent_Chld_Render", Rend_Chld, SOFM_Rend);
        Dependency("Port_Attach_Down_Stream", SIML_SOFM, SOFM_Rend, 3);
        Dependency("Port_Attach_Down_Stream", Common_Grid, SIML_SOFM);
        }
}
```

Listing A.5: Instance Collections in Lua

```
App_INST =
{
["_TABL_Name"] = "Instances_01",
["Coll_Parent_Child1"] =  {["_User_Note"] ="Host_Window_and_Child_Grid_Control",
        ["_Coll_List"] = {-- (c) Instances
                ["Master_Wnd"]=App_DEFN._COMP_GUI.Host_Wnd,
                ["Common_Grid"]=App_DEFN._COMP_GUI.Chld_Wnd_Lft,
                ["Rend_Chld"]=App_DEFN._COMP_GUI.Chld_Wnd_Rgt,
                ["SOFM_Rend"]=App_DEFN._COMP_GUI.OGL_2D_Parts,
                ["SIML_SOFM"]=App_DEFN._COMP_FNC.Simul_Sofm,
                },
        ["_Dpnd_PreP"] = { -- (d) Dependencies
                {["_Dpnd_Type"] = {"_Parent_Child_Wnd",},
                ["_Dpnd_List"] = {"Master_Wnd","Common_Grid","Rend_Chld",},
                ["_Dpnd_Maps"] = {
                        {["_Dpnd_Map_Type"]="_Wnd_Arrangement",
                        ["_Dpnd_Map_Arrangement"]="WndArr1",
                        ["_Dpnd_Map_Layout_Name"]="Default",
                        ["_Dpnd_Map_List"]={
                                {["_Name"]="a",["_Comp"]="Common_Grid",},
                                {["_Name"]="b",["_Comp"]="Rend_Chld",},
                                },
                        },
                },
                {["_Dpnd_Type"]={"_Parent_Child_Render",},
                ["_Dpnd_List"]={"Master_Wnd","Rend_Chld","SOFM_Rend",},
                        },
                {["_Dpnd_Type"]={"_Port_Attach_Down_Stream",},
                ["_Dpnd_List"]={"SIML_SOFM","SOFM_Rend","SOFM_Rend","SOFM_Rend",},
                        },
                {["_Dpnd_Type"]={"_Port_Attach_Down_Stream",},
                ["_Dpnd_List"]={"Common_Grid","SIML_SOFM",},
                        },
                },
        },
}
```

A.2.5 Application Table

The application table is the simplest part of the meta-program. We are simply agglomerating the alias and instance tables in two lists. The application table essentially defines the program application in terms of the corresponding alias and instance tables.

Listing A.6: Application Definition in Pseudocode

```
Application App_01
{
List_Aliases(Types_01);
List_Instances(Instances_01);
}
```

Listing A.7: Application Definition in Lua

```
App_TABL =
{
["_TABL_Name"] = "App_01",
["_App_DEFN"] = { Types_01, }, -- A List of all component atom tables.
["_App_INST"] = { Instances_01, }, -- A List of all component instance tables.
}
```

A.3 Component Communication

Communication between components takes place via *component ports* (not to be confused with internet ports). Ports are mediated by the use of memory mapped files, mutex delimited access to these files and event signalling to coordinate between the transmitter and receiver. We refer to the transmitter as the *author* and the receiver as the *editor*. Despite this terminology ports are symmetric in structure offering bi-directional communications. In cases where the data flow

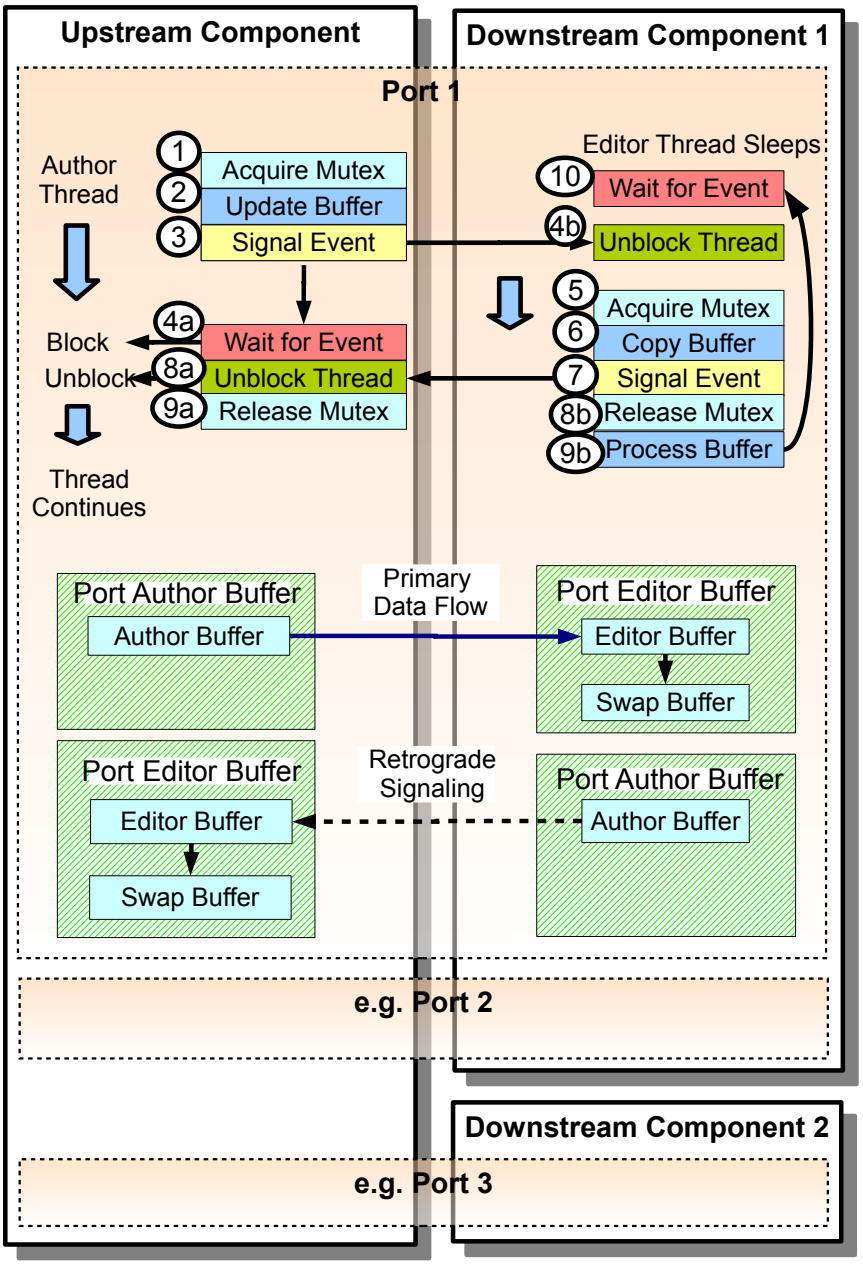


Figure A.3: Component Ports for Communication

is primariliy in one direction we sometimes refer to the transmitting component as *upstream* and the receiving component as *downstream*. In this case if the downstream component is sending a signal to the upstream component we refer to this as *retrograde* signalling.

The essential structure of a port is illustrated in figure A.3. There are nine steps in a typical port interaction;

1. The upstream component first acquires the mutex that controls access to the *author* buffer. Note that the author buffer is a named memory mapped file. As part of the meta-program initialization process all ports are created in a fashion that each component is passed details of the component to which it is coupled.

2. The upstream buffer then writes to the author buffer.

3. Once this has been done a named event is signalled indicating to the receiving component that the buffer has been modified.

4. The thread in the upstream component will then enter a blocked state. The named event signalled in the upstream component releases the *editor* thread in the downstream component from a blocked state.

5. The editor thread in the receiving component then acquires a mutex guarding the editor buffer. Note that this *editor* buffer is the same memory mapped file as that which corresponds to the *author* buffer. During component initialization the sngle name that corresponds to this memory mapped file is passed to both the upstream and downstream components so that they both share access to this memory mapped file.

6. The *editor* thread then copies the contents of the *editor* buffer to a *swap* buffer. Note that the editor buffer is the same memory mapped file as the author buffer, so we are effectively just copying the contents of the author buffer (controlled by the upstream component) into buffer controlled by the downstream component.

7. Once it has finished copying the editor buffer then signals a named event.

8. This named event becoming signalled then releases the blocked *author* thread. The editor thread continues to release the mutex guarding access to the *editor* buffer.

9. The author thread also releases the mutex controlling access to the author buffer. The editor thread carries out any additional processing that is required by the downstream component.

10. Once the *editor* thread has completed processing the newly arrived data it returns its dormant blocked state.

This example illustrates how asynchronous communication takes place between differing components. There is also support for synchronous communication where prior to the final step 10, the editor thread indicates that all processing has been completed through signalling an additional event. Depending upon the message type sent by the author thread, it will either continue without waiting for this event to become signalled (aysnchronous mode) or await the signalling of this event (synchronous mode).

Figure A.2 shows what ports are constructed in support of the minimal simulation application illustrated in figure A.4.

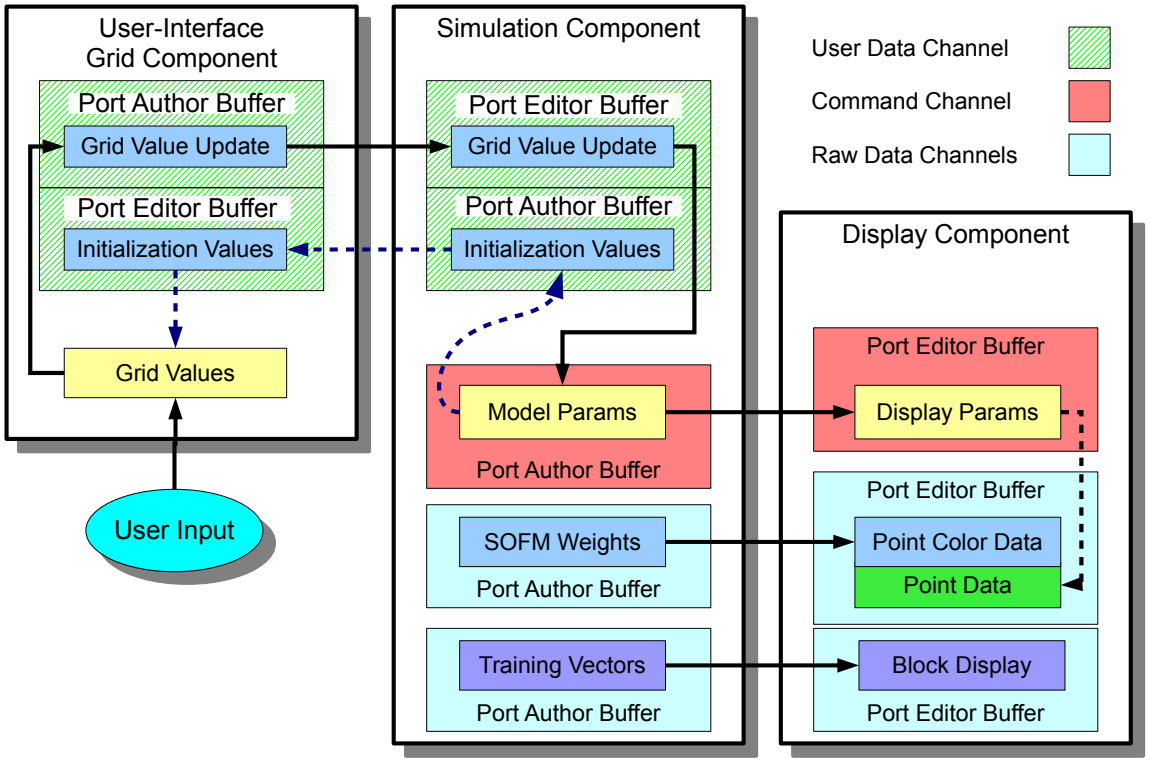


Figure A.4: Minimal Simulation Components

References

Evolutionary Robotics: Exploring New Horizons. *New Horizons in Evolutionary Robotics*, pages 3–25, 2011. 70

Sophie Achard and Ed Bullmore. Efficiency and cost of economical brain functional networks. *PLoS computational biology*, 3(2):e17, February 2007. 219

Kenneth Aizawa. Connectionism and artificial intelligence : history and philosophical interpretation. *Journal of Experimental & Theoretical Artificial Intelligence*, 4:295–313, 1992. 39

R.L. Albin, A.B. Young, and J.B. Penney. The functional anatomy of basal ganglia disorders. *Trends in neurosciences*, 12(10):366–375, 1989. 173

G E Alexander, M D Crutcher, and Michael D. Crutcher Garret E. Alexander. Functional architecture of basal ganglia circuits: neural substrates of parallel processing. *Trends in neurosciences*, 13(7):266–71, July 1990. 173

R D Alexander. The evolution of social behavior. *Annual Review of Ecology and Systematics*, 5(1974):325–383, 1974. 26

R. McNeill Alexander. *Principles of Animal Locomotion*. Princeton University Press, Princeton (NJ), 2003. 143, 152, 219

Adelbert Ames. CNS energy metabolism as related to function. *Brain research. Brain research reviews*, 34(1-2):42–68, November 2000. 221

Michael L. Anderson. Embodied Cognition: A field guide. *Artificial Intelligence*, 149(1):91–130, September 2003. 57

Peter J Angeline and Jordan B Pollack. Competitive Environments Evolve Better Solutions for Complex Tasks. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 264 – 270, San Francisco, CA, 1993. Morgan Kaufmann Publishers Inc. 77

Peter J Angeline and Jordan B Pollack. Coevolving High-Level Representations (Santa Fe Institute Series). In *Artificial Life III*. Westview Press, 1994. 77

Peter J Angeline, Gregory M Saunders, and Jordan B Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 5(1): 54–65, January 1994. 77

Ronald C Arkin. *Behavior-Based Robotics*, volume 2. MIT Press, 1998. 55, 57

W. Ross Ashby. *Design for a Brain - The origin of adaptive behavior*. John Wiley & Sons, 1960. 44, 45, 47, 48, 49, 91, 93, 94, 95, 96, 313

WR Ashby. *An introduction to cybernetics*. Chapman and Hall Ltd, London, 1956. 44

J C Astor and C Adami. A developmental model for the evolution of artificial neural networks. *Artificial life*, 6(3):189–218, January 2000. 36

D Attwell and S B Laughlin. An energy budget for signaling in the grey matter of the brain. *Journal of cerebral blood flow and metabolism : official journal of the International Society of Cerebral Blood Flow and Metabolism*, 21(10): 1133–45, October 2001. 221

JE Auerbach and JC Bongard. Evolving complete robots with CPPN-NEAT: the utility of recurrent connections. In *Genetic and Evolutionary Computation Conference (GECCO 2011)*, Dublin, IR, 2011. 79

Joshua E. Auerbach and Josh C. Bongard. Evolving CPPNs to grow three-dimensional physical structures. *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO '10*, page 627, 2010. 79

Jason M. Baker. Adaptive speciation: the role of natural selection in mechanisms of geographic and non-geographic speciation. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 36(2):303–326, June 2005. 37

Karthik Balakrishan and Vasant Honavar. Evolutionary Design of Neural Architectures - A Preliminary Taxonomy and Guide to Literature. Technical report, Artifciail Intelligence Research Group. Iowa State University, Ames, Iowa, 1995. 66

Gianluca Baldassarre and Stefano Nolfi. Strengths and Synergies of Evolved and Designed Controllers: A Study within Collective Robotics. *Artificial Intelligence*, 173(7-8):857–875, 2007. 74

Gianluca Baldassarre, Stefano Nolfi, and Domenico Parisi. Evolving mobile robots

able to display collective behaviors. *Artificial life*, 9(3):255–67, January 2003a. 29, 74

Gianluca Baldassarre, Stefano Nolfi, and Domenico Parisi. Evolving mobile robots able to display collective behaviors. *Artificial life*, 9(3):255–67, January 2003b. 74

Gianluca Baldassarre, Vito Trianni, Michael Bonani, Francesco Mondada, Marco Dorigo, and Stefano Nolfi. Self-organized coordinated motion in groups of physically connected robots. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 37(1):224–39, February 2007. 74

J M Baldwin. A new factor in evolution. *The American Naturalist*, 30(354): 441–451, 1896. 34

Moshe Bar. Predictions: a universal principle in the operation of the human brain. Introduction. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1521):1181–2, May 2009. 213

Izhar Bar-Gad, Genela Morris, and Hagai Bergman. Information processing, dimensionality reduction and reinforcement learning in the basal ganglia. *Progress in neurobiology*, 71(6):439–73, December 2003. 172

J S F Barker. Simulation of genetic systems by automatic digital computers : IV Selection between Allelles at a Sex Linked Locus. *Australian J. Biological Sciences*, 11(4):613–626, 1958a. 30

J S F Barker. Simulation of genetic systems by automatic digital computers :

III Slection Between Alleles at an Autosomal Locus. *Australian J. Biological Sciences*, 11(4):603–612, 1958b. 30

Lionel Barnett. Netcrawling - Optimal Evolutionary Search with Neutral Networks. *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, pages 30–37, 2000. 36

N.A. Barricelli. "esempi numerici di processi di evoluzione". *Methodos*, pages 45–68, 1954. 29

N.A. Barricelli. "symbiogenetic evolution processes realized by artificial methods". *Methodos*, 9:143–182, 1957. 29

NA Barricelli. Numerical testing of evolution theories. *Acta Biotheoretica*, 1962. 29, 30

William Bechtel and Adele Abrahamsen. Connectionist Architectures. In *Connectionism and the mind*, pages 21–65. Basil Blackwell, 1991. 39

B G Beddall. *Journal of the history of biology*, 5(1):153–8, January 1972. 25

R. D. Beer. On the Dynamics of Small Continuous-Time Recurrent Neural Networks. *Adaptive Behavior*, 3(4):469–509, March 1995a. 76

R. D. Beer and J. C. Gallagher. Evolving Dynamical Neural Networks for Adaptive Behavior. *Adaptive Behavior*, 1(1):91–122, June 1992a. 68, 69

Randall D Beer. Dynamical Systems and Embedded Cognition. In K. Frankish and W. Ramsey, editors, *The Cambridge Handbook of Artificial Intelligence*, number 812, pages 856–873. Cambridge University Press, a. 76

Randall D Beer. Dynamical Analysis of Evolved Agents : A Primer. In P. Vargas, E. Di Paolo, I. Harvey, and P. Husbands, editors, *The Horizons for Evolutionary Robotics*, number 812, pages 856–873. b. 76

Randall D. Beer. *Intelligence as adaptive behavior: an experiment in computational neuroethology*. Academic Press Professional, Inc., San Diego, CA, USA, 1990. 54, 69, 75

Randall D. Beer. A dynamical systems perspective on autonomous agents (Technical Report CES-92-11). Technical report, Dept. of Computer Engineering and Science, Case Western Reserve University, 1992. 76

Randall D Beer. A dynamical systems perspective on agent-environment interaction. *Artificial intelligence*, 72:173–215, 1995b. 105, 115, 116

Randall D. Beer. The dynamics of adaptive behavior: A research program. *Robotics and Autonomous Systems*, 20:257–289, 1997. 76

Randall D Beer. Framing the debate between computational and dynamical approaches to cognitive science (A commentary on "The dynamical hypothesis in cognitive science" by Tim van Gelder). *Behavioral and Brain Sciences*, 21 (5):630, 1998. 76

Randall D. Beer. Autopoiesis and Cognition in the Game of Life. *Artificial Life*, 10(3):309–326, June 2004. 76

Randall D Beer. Parameter Space Structure of Continuous-Time Recurrent Neural Networks. *Neural Computation*, 18:3009–3051, 2006. 76

Randall D. Beer. Computer Evolution of Chemotaxis in Model Nematodes. *Brain, Behavior and Evolution*, 77(1):1–2, 2011. 76

Randall D. Beer and J. C. Gallagher. *Evolving Dynamical Neural Networks for Adaptive Behavior*, volume 1. June 1992b. 76

R.D. Beer. Toward the evolution of dynamical neural networks for minimally cognitive behavior. In Pattie Maes, Maja J. Mataric, Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson, editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (Complex Adaptive Systems)*, volume 4, pages 421–429, 1996. 76, 101

Rd Beer. Dynamical approaches to cognitive science. *Trends in cognitive sciences*, 4(3):91–99, March 2000. 76

RD Beer. The dynamics of brain-body-environment systems: A status report. *Handbook of cognitive science: An embodied*, pages 99–120, 2008. 76

Richard K Belew, John McInerney, and Nicol N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. Technical report, Cognitive Computer Science Research Group, Univ. California at San Diego, 1990. 66

R.E. Bellman. *Dynamic Programming)*. Princeton University Press, 1957. 187, 211

Kent C Berridge. The debate over dopamine's role in reward: the case for incentive salience. *Psychopharmacology*, 191(3):391–431, April 2007. 179

D.P. Bertsekas and J.N. Tsitsiklis. *Neurodynamic Programming (Optimization and Neural Computation Series, 3)*. Athena Scientific, 1996. 189, 212

Hans-georg Beyer and Hans-paul Schwefel. Evolution strategies. *Natural Computing*, 1:3–52, 2002. 32

A. Billard and A. K. Ijspeert. Biologically Inspired Neural Controllers for Motor Control in a Quadruped Robot. In S. Amari, C.L. Giles, M. Gori, and V. Piuri, editors, *IEEE-INNS-ENNS International JointConference on Neural Networks (IJCNN 2000)*, volume 6, pages 6637–6641, 2000. 102

Christopher M Bishop. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006. 38

Anders Björklund and Stephen B Dunnett. Dopamine neuron systems in the brain: an update. *Trends in neurosciences*, 30(5):194–202, May 2007. 178, 179

J. Blynel and D. Floreano. Levels of Dynamic and Adaptive Behaviorin Evolutionary Neural Controllers. In B. Hallam, D. Floreano, J. Hallam, G. Hayes, and J.-A. Meyer, editors, *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior (SAB 02)*, pages 272–281. MIT Press, 2002. 111, 119

Jesper Blynel. Evolving reinforcement learning-like abilities for robots. *Evolvable Systems: From Biology to Hardware*, 2003. 292

Rafal Bogacz, M.W. Brown, and C. Giraud-Carrier. Model of co-operation between recency, familiarity and novelty neurons in the perirhinal cortex. *Neurocomputing*, 38:1121–1126, 2001. 186, 197

Daniel I. Bolnick and Benjamin M. Fitzpatrick. Sympatric Speciation: Models and Empirical Evidence. *Annual Review of Ecology, Evolution, and Systematics*, 38(1):459–487, December 2007. 37

J.C. Bongard and C. Paul. Making Evolution an Offer it Can't Refuse: Morphology and the Extradimensional Bypass. In J. Keleman and P. Sosik, editors, *Proceedings of the Sixth European Conference on Artificial Life (ECAL 2000)*, pages 401–412, 2001. 102

JC Bongard and Rolf Pfeifer. Evolving complete agents using artificial ontogeny. In F. Hara and Rolf Pfeifer, editors, *Morpho-functional machines: the new species : designing embodied intelligence*, chapter 13, pages 237–258. Springer, 2003. 36, 69

Josh Bongard. Evolving Dynamic Gaits on a Physical Robot. In M. Keijzer, editor, *Proceedings of Genetic and Evolutionary Computation Conference, Late Breaking Paper*, Seattle, Washington, USA, 2004. 78

Josh Bongard, Victor Zykov, and Hod Lipson. Resilient machines through continuous self-modeling. *Science (New York, N.Y.)*, 314(5802):1118–21, November 2006. 79

Josh C Bongard. *Incremental Approaches to the Combined Evolution of a Robot s Body and Brain*. Phd, University of Zurich, 2003. 79

Josh C. Bongard. A probabilistic functional crossover operator for genetic programming. *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO '10*, page 925, 2010. 79

Josh C. Bongard. Innocent Until Proven Guilty: Reducing Robot Shaping From Polynomial to Linear Time. *IEEE Transactions on Evolutionary Computation*, 15(4):571–585, August 2011. 80

Josh C. Bongard and Gregory S. Hornby. Guarding against premature convergence while accelerating evolutionary search. *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO '10*, page 111, 2010. 79

Anna M. Borghi, Domenico Parisi, and Andrea di Ferdinando. Action and hierarchical levels of categories: A connectionist perspective. *Cognitive Systems Research*, 6(2):99–110, June 2005. 74

DH Boucher and Sam James. The ecology of mutualism. *Annual Review of Ecology and Systematics*, 13:315–347, 1982. 28

Tiago Branco and Kevin Staras. The probability of neurotransmitter release: variability and feedback control at single synapses. *Nature reviews. Neuroscience*, 10(5):373–83, May 2009. 233

H J Bremermann. OPTIMIZATION THROUGH EVOLUTION AND RECOMBINATI ON Part I . Limitations On Data Processing Arising From Quantum Theory. In M.C. Yovits, G.T. Jacobi, and G.D. Goldstein, editors, *Self-Organizing Systems*, pages 93–106. Spartan Books, Washington DC, 1962. 31

F.C. Brentano and L.L. McAlister. *Psychology from an empirical standpoint*. Psychology Press, 1995. 56

C Bresch, U Niesert, and D Harnasch. Hypercycles, parasites and packages. *Journal of theoretical biology*, 85(3):399–405, August 1980. 28

Ethan S. Bromberg-Martin, Masayuki Matsumoto, and Okihide Hikosaka. Dopamine in Motivational Control: Rewarding, Aversive, and Alerting. *Neuron*, 68(5):815–834, December 2010. 179

Daniel R Brooks. The Mastodon in the room: how Darwinian is neo-Darwinism? *Studies in history and philosophy of biological and biomedical sciences*, 42(1): 82–8, March 2011. 28

Rodney A Brooks. A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network (A.I. Memo 1091). Technical report, MASSACHUSETTS INSTITUTE OF TECHNOLOGY ARTIFICIAL INTELLIGENCE LABORATORY, 1989. 55

Rodney A Brooks. Elephants don't play chess. *Robotics and autonomous systems*, 6, 1990. 51, 57

Rodney A Brooks. Intelligence without representation. *Artificial intelligence*, 47 (1811):139–159, 1991. 43, 51, 53, 54, 57

Rodney A Brooks. Artificial Life and Real Robots. In Francisco J. Varela and Paul Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 3–10, Cambridge, MA, 1992. MIT Press. 55

J.S. Bruner and C.C. Goodman. Value and need as organizing factors in perception. *The journal of abnormal and social psychology*, 42(1):33, 1947. 56

J.S. Bruner and L. Postman. Symbolic value as an organizing factor in perception. *The Journal of social psychology*, 27(2):203–208, 1948. 56

C Buhusi. Timing in simple conditioning and occasion setting: a neural network approach. *Behavioural Processes*, 45(1-3):33–57, April 1999. 186

P.-L Buono and M. Golubitsky. Models of central pattern generators for quadruped locomotion. *Journal of Mathematical Biology*, 42:291–326, 2001. 140

Timothy J Buschman and Earl K Miller. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *Science (New York, N.Y.)*, 315(5820):1860–2, March 2007. 340

Daniel Bush, Andrew Philippides, Phil Husbands, and Michael O'Shea. Spike-timing dependent plasticity and the cognitive map. *Frontiers in computational neuroscience*, 4(October):142, January 2010. 73

G. Buzsáki, K. Kaila, and M. Raichle. Inhibition and Brain Work. *Neuron*, 56 (5):771–783, February 2007. 222

Roberto Cabeza, Elisa Ciaramelli, Ingrid R Olson, and Morris Moscovitch. Parietal Cortex and Episodic Memory: An Attentional Account. *Nature Reviews Neuroscience*, 9(8):613–625, 2009. 340

Roberto Cabeza, Elisa Ciaramelli, and Morris Moscovitch. Cognitive contributions of the ventral parietal cortex: an integrative theoretical account. *Trends in cognitive sciences*, 16(6):338–52, June 2012. 340

R. Calabretta and D Parisi. *"Evolutionary Connectionism and Mind/Brain Modularity"*, chapter 14, pages 309–330. "The Vienna Series in Theoretical Biology". MIT Press, 2005. 36

Raffaele Calabretta, Stefano Nolfi, Domenico Parisi, and Günter P Wagner. Duplication of modules facilitates the evolution of functional specialization (Technical Report 59 Yale Center for Computational Ecology). Technical report, Neural Systems and Artificial Life Group, Institute of Psychology, National Research Council, Rome, 2000. 74

Á. Campo and J. Santos. Evolution of adaptive center-crossing continuous time recurrent neural networks for biped robot control. In *ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning*, number April, pages 535–540, 2010. 141

Angelo Cangelosi and Jeffrey L. Elman. Gene regulation and biological development in neural networks: an exploratory model (Technical Report No. CRL-UCSD). Technical report, University of California at San Diego, 1995. 70

Angelo Cangelosi and Stefano Nolfi. Artificial life models of neural development. In Sanjeev Kumar and Peter J. Bentley, editors, *On Growth, Form and Computers*, pages 339–352. Academic Press, 1st edition, 2003. 37

Angelo Cangelosi and Domenico Parisi. The Emergence of a 'Language' in an Evolving Population of Neural Networks. *Connection Science*, 10(2):83–97, June 1998. 74, 77

Angelo Cangelosi, Domenico Parisi, and Stefano Nolfi. Cell division and migration

in a'genotype'for neural networks. *Network: Computation in Neural Systems*, 1994. 36

Genci Capi and Kenji Doya. Evolution of Neural Architecture Fitting Environmental Dynamics. *Adaptive Behaviour*, 13:53–66, 2005. 290, 340

Andrew Caplin and Mark Dean. Dopamine, Reward Prediction Error, and Economics. *Quarterly Journal of Economics*, 123(2):663–701, 2008. 180

Peter a. Cariani. The homeostat as embodiment of adaptive control. *International Journal of General Systems*, 38(2):139–154, February 2009. 47, 51

F. Cecconi, D. Denaro, D. Parisi, and U Piazzalunga. Social aggregations in evolving neural networks. In C Castelfranchi and E Werner, editors, *Artificial social systems. Proceedings of MAAAMAW 92, 4th European Workshop on Modelling Autonomous Agents in a Multi Agents World*, pages 41–54, Berlin, Heidelberg, 1994. Springer-Verlag. 73

V S Chakravarthy, Denny Joseph, and Raju S Bapi. What do the basal ganglia do? A modeling perspective. *Biological cybernetics*, 103(3):237–53, September 2010. 181

David J Chalmers. The Evolution of Learning : An Experiment in Genetic Connectionism. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*, San Mateo, CA, 1990. Morgan Kaufmann. 35

Alastair Channon and Bob Damper. The Artificial Evolution of Real Intelligence

by Natural Selection. In *European Conference on Artificial Life (ECAL97)*, Brighton, 1997. 70

Peter Chesson. Mechanisms of maintenance of species diversity. *Annual Review of Ecology and Systematics*, pages 343–366, 2000. 37

H J Chiel and R D Beer. Simulation of adaptive behavior. *Current opinion in neurobiology*, 1(4):605–9, December 1991. 76

Hillel J. Chiel and Randall D. Beer. The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends in Neurosciences*, 20(12):553–557, December 1997. 76

Hillel J. Chiel and Randall D. Beer. Computational neuroethology. In Larry R. Squire, editor, *Encyclopedia of the Neuroscience*. Elsevier, 2008. 54

Alonzo Church and Alan M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *The Journal of Symbolic Logic*, 2(1):42, March 1937. 46

Andrew Clark. *Being There: Putting Brain Body and World Together Again*. MIT, 1997. 56

D. Cliff, I. Harvey, and P. Husbands. Incremental Evolution of Neural Network Architectures for Adaptive Behaviour. Technical Report December, CSRP 256, University of Sussex, School of Cognitive and Computer Science, Brighton, 1992. 69, 72

D. Cliff, Inman Harvey, and Phil Husbands. Artificial evolution of visual control systems for robots, 1997. 72

D. T. Cliff, I. Harvey, and P. Husbands. Explorations in Evolutionary Robotics. *Adaptive Behaviour*, 2(1):71–108, 1993a. 69, 72

Dave Cliff. Computational neuroethology: a provisional manifesto. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 29–39, Cambridge, MA, USA, 1990. MIT Press. 54, 69

Dave Cliff. Neuroethology, Computational. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks: Second Edition*, pages 737–740. A Bradford Book, 2002. 54

Dave Cliff, Philip Husbands, and Inman Harvey. Evolving Visually Guided Robots. In Jean-arcady Meyer, Herbert L. Roitblat, and Stewart W. Wilson, editors, *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Be- havior*, number July 1992, pages 374–383. A Bradford Book, 1993b. 69, 72

Jeff Clune and Hod Lipson. Evolving Three-Dimensional Objects with a Generative Encoding Inspired by Developmental Biology. In Tom Lenaerts, Mario Giacobini, Hugues Bersini, Paul Bourgine, Marco Dorigo, and René Doursat, editors, *Advances in Artificial Life, ECAL 2011 Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 141–148. MIT Press, 2011. 78

Jeff Clune, Benjamin E. Beckmann, Charles Ofria, and Robert T. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative en-

coding. *2009 IEEE Congress on Evolutionary Computation*, pages 2764–2771, May 2009a. 82

Jeff Clune, Charles Ofria, and Robert T. Pennock. The sensitivity of HyperNEAT to different geometric representations of a problem. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO '09*, page 675, 2009b. 82

Michael X M.X. Cohen and M.J. Michael J Frank. Neurocomputational models of basal ganglia function in learning, memory and choice. *Behavioural brain research*, 199(1):141–156, April 2009. 179, 181

M Conrad. Evolutionary learning circuits. *Journal of theoretical biology*, 46(1): 167–88, July 1974. 67

Michael Conrad, Roberto R Kampfner, Kevin G Kirby, Elizabeth N Rizki, George Schleis, Robert Smalz, and Robert Trenary. Towards an artificial brain. *BioSystems*, 23:175–218, 1989. 67

Michael J Constantino, Diane B Arnkoff, Carol R Glass, Rebecca M Ametrano, and Julianna Z Smith. Neural repetition suppression reflects fulfilled perceptual expectations. *Nature Neurorscience*, 11(9):1004–1006, December 2008. 220

Roshan Cools. Dopaminergic control of the striatum for high-level cognition. *Current opinion in neurobiology*, 21(3):402–407, April 2011. 179

Jack B. Copeland and Diane Proudfoot. On Alan Turing's Anticipation of Connectionism. *Synthese*, 108(3):361–377, 1996. 47

Maurizio Corbetta and Gordon L Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews. Neuroscience*, 3(3):201–15, March 2002. 340

Maurizio Corbetta, Gaurav Patel, and Gordon L. Shulman. The Reorienting System of the Human Brain: From Environment to Theory of Mind. *Neuron*, 58(3):306–324, 2008. 340

Roberto Cordeschi. Early-Connectionism Machines. *AI & Society*, 14:314–330, 2000. 39

J D Cowan. DISCUSSION: McCULLOCH-PITTS AND RELATED NEURAL NETS FROM 1943 TO 1989. *Bulletin of mathematical biology*, 52(1-2):73–97, 1990. 39

Jack D Cowan and David H Sharp. Neural Nets and Artificial Intelligence. *Daedalus*, 117(1):85–121, 2012. 39

Jerry A. Coyne and H. Allen Orr. *Speciation*. Sinauer Associates, 2004. 37

Frederick L. Crabbe and Michael G. Dyer. *Advances in the Evolutionary Synthesis of Intelligent Agents*, chapter Goal Directed Adaptive Behaviour in Second-Order Neural Netwworks: Learning and Evolving in the MAXSON Architecture, pages 307–335. MIT Press, 2001a. 291

Frederick L Crabbe and Michael G Dyer. Goal Directed Adaptive Behavior in Second-Order Neural Networks : The MAXSON family of architectures. *Adaptive Behavior*, 8(2):145–168, 2001b. 291

NL Cramer. A representation for the adaptive generation of simple sequential programs. *Proceedings of an International Conference on Genetic Algorithms and the Applications*, pages 183–187, 1985. 32

Francis Crick. The recent excitement about neural networks. *Nature*, 337(12 January):129–132, 1989. 43

K Dale and T S Collett. Using artificial evolution and selection to model insect navigation. *Current Biology*, 11(17):1305–16, September 2001. 73

Kyran Dale and Phil Husbands. The evolution of reaction-diffusion controllers for minimally cognitive agents. *Artificial life*, 16(1):1–19, January 2010. 73, 141

Yang Dan and M.M. Poo. Spike timing-dependent plasticity: from synapse to perception. *Physiological reviews*, 86(3):1033, 2006. 238

Yang Dan and Mu-Ming Poo. Spike timing-dependent plasticity of neural circuits. *Neuron*, 44(1):23–30, September 2004. 238

Charles Darwin. *On the Origin of Species*, volume 5. John Murray, 1859. 23

Dipankar Dasgupta and D.R. McGregor. Designing application-specific neural networks using the structured genetic algorithm. In *[Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks*, number 2, pages 87–96. IEEE Comput. Soc. Press. 66

J Davison. Genetic exchange between bacteria in the environment. *Plasmid*, 42 (2):73–91, September 1999. 28

Nathaniel D Daw, Aaron C Courville, David S Tourtezky, and David S Touretzky. Representation and timing in theories of the dopamine system. *Neural computation*, 18(7):1637–77, July 2006. 179

R Dawkins. *The Extended Phenotype*. Oxford University Press, 1982. 26, 27

Richard Dawkins. *The Selfish Gene*, volume 32. Oxford University Press, 1976. 25

Hugo de Garis. Building Artificial Nervous Systems with Genetically Programmed Neural Network Modules. *Proceedings of the 7th International Conference on Machine Learning*, pages 132–139, 1990. 68

Kenneth De Jong. Genetic algorithms: A 10 year perspective. *First International Conference on Genetic Algorithms*, 1985. 32

Gonzalo G de Polavieja. Errors drive the evolution of biological signalling to costly codes. *Journal of theoretical biology*, 214(4):657–64, February 2002. 220

Suash Deb. Why Computers Are Not Displaying Intelligence. *2008 International Conference on Advanced Computer Theory and Engineering*, pages 44–49, December 2008. 51

Daniele Denaro and Domenico Parisi. Cultural Evolution in a Population of Neural Networks. In *Proceedings of the 8th Italian Workshop on Neural Nets*, volume 12, pages 100–111, Vietri sul Mare, 1997. 74

M. Denham. *Emergent neural computational architectures based on neuroscience: towards neuroscience-inspired computing*, chapter The dynamics of learning

and memory: lessons from neuroscience, pages 333 – 347. Lecture Notes in Computer Science. Springer, 2001. 240

Ezequiel A. Di Paolo. Spike-timing dependent plasticity for evolved robots. *Adaptive Behavior*, 10(3-4):243–263, 2002. 73

Ezequiel A. Di Paolo. Organismically-inspired robotics : homeostatic adaptation and teleology beyond the closed sensorimotor loop. In K. Asakura and T. Murase, editors, *Dynamical Systems Approach to Embodiment and Sociality*, pages 19–42. Adelaide, 2003a. 73

Ezequiel A. Di Paolo. Evolving spike-timing-dependent plasticity for single-trial learning in robots. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 361(1811):2299–319, October 2003b. 73

Ezequiel A. Di Paolo. Autopoiesis, Adaptivity, Teleology, Agency. *Phenomenology and the Cognitive Sciences*, 4(4):429–452, April 2006. 73

Ezequiel A. Di Paolo. Extended Life. *Topoi*, 28(1):9–21, December 2008. 73

Ezequiel A Di Paolo. Robotics Inspired in the Organism The meaningless machine. *Intellectica*, 1-2(53-54):129–162, 2010. 73

Ezequiel a Di Paolo and Hiroyuki Iizuka. How (not) to model autonomous behaviour. *Bio Systems*, 91(2):409–23, February 2008. 73

Mehrdad Dianati, Insop Song, and Mark Treiber. An Introduction to Genetic Algorithms and Evolution Strategies. 33

U. Dieckmann, M Doebeli, Metz. J.A.J., and D Tautz, editors. *"Adaptive Speciation"*. "Cambridge Studies in Adaptive Dynamics". "Cambride University Press", 1 edition, 2004. 37

Jun Ding, Jayms D Peterson, and D James Surmeier. Corticostriatal and thalamostriatal synapses have distinctive properties. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 28(25):6483–92, June 2008. 233, 300

Peter F Dominey. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological Cybernetics*, 274: 265–274, 1995. 291

Jack Dongarra. Trends in high - performance computing. *IEEE Circuits and Devices Magazine*, 22(1):22–27, January 2006. 51

Marco Dorigo, Vito Trianni, S Erol, Roderich Groß, Thomas H Labella, Gianluca Baldassarre, Stefano Nolfi, Jean-louis Deneubourg, Francesco Mondada, Dario Floreano, and Luca M Gambardella. Evolving Self-Organizing Behaviors for a Swarm-bot. *Autonomous Robots*, 17(2-3):223–245, 2004. 74

Kenji Doya. Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology*, 10:732–739, 2000. xvi, xxxvi, 340, 344, 345

Kenji Doya. Modulators of decision making. *Nature neuroscience*, 11(4):410–6, April 2008. 179

Julien Doyon, Pierre Bellec, Rhonda Amsel, Virginia Penhune, Oury Monchi,

Julie Carrier, Stéphane Lehéricy, and Habib Benali. Contributions of the basal ganglia and functionally related brain structures to motor learning. *Behavioural brain research*, 199(1):61–75, April 2009. 172

W.B. Dress. Darwinian optimization of synthetic neural systems. In *IEEE international conference on neural networks*, San Diego, CA, 1987. Oak Ridge National Lab. 65, 66

W.B. Dress and J.R. Knisley. A darwinian approach to artificial neural systems. In *IEEE international conference on systems, man, and cybernetics*. Oak Ridge National Lab, 1987. 65

Hubert L. Dreyfus. Computers Must Have Bodies in order to be Intelligent. *Review of Metaphysics*, 21(1):13–32, 1967. 50, 51, 52, 59

Hubert L. Dreyfus. Why Heideggerian AI Failed and How Fixing it Would Require Making it More Heideggerian. *Philosophical Psychology*, 20(2):247–268, April 2007. 59, 62, 63

P. Durr, Claudio Mattiussi, and Dario Floreano. Genetic representation and evolvability of modular neural controllers. *Computational Intelligence Magazine, IEEE*, 5(3):10–19, 2010. 37

D Durstewitz, M Kelc, and O Güntürkün. A neurocomputational theory of the dopaminergic modulation of working memory functions. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 19(7):2807–22, April 1999. 179

Marc Ebner, Mark Shackleton, and Rob Shipman. How neutral networks influence evolvability. *Complexity*, 7(2):19–33, November 2001. 36

J C Eccles. The neurophysiological basis of mind: The principles of neurophysiology. *Clarendon Press Oxford*, page 209, 1953. 40

G Edelman. *Neural Darwinism: The theory of Neuronal Group Selection*. Basic Books, Inc., 1 edition, 1987. 67

M. D. Egbert and E. Di Paolo. Integrating Autopoiesis and Behavior: An Exploration in Computational Chemo-ethology. *Adaptive Behavior*, 17(5):387–401, September 2009. 73

Peter Eggenberger. Cell interactions as a control tool of developmental processes for evolutionary robotics. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 440–448, 1996. 36, 70

Peter Eggenberger. Evolving morphologies of simulated 3d organisms based on differential gene expression. *Proc of the Fourth European Conference on Artificial Intelligence*, 1997a. 70

Peter Eggenberger. Creation of neural networks based on developmental and evolutionary principles. In *In Proceedings of the International Conference on Artificial Neural Networks ICANN'97*, 1997b. 36, 38, 70

A. E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, 2010. 32

Howard Eichenbaum and Norbert J Fortin. The neurobiology of memory based

predictions. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1521):1183–91, May 2009. 339

J.G. Elias. Genetic generation of connection patterns for a dynamic artificial neural network. In *[Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pages 38–54, Baltimore, MD, 1992. IEEE Comput. Soc. Press. 66

J Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, June 1990. 290

Jihua Feng and Wenjuan Li. A Hybrid Genetic Algorithm with Fitness Sharing Based on Rough Sets Theory. In *2006 6th World Congress on Intelligent Control and Automation*, pages 3340–3344. IEEE, 2006. 81

Andrea Di Ferdinando, Raffaele Calabretta, and Domenico Parisi. Evolving Modular Architectures for Neural Networks. Technical report, Neural Systems and Artificial Life Group, Institute of Psychology, National Research Council, Rome, 2000. 74

Sevan G Ficici, Richard A Watson, and Jordan B Pollack. Embodied Evolution: A Response to Challenges in Evolutionary Robotics. In J.L. Wyatt and J. Demiris, editors, *Proceedings of the Eighth European Workshop on Learning Robots,*, pages 14–22, Lausanne, Switzerland, 1999. 77

D Filliat and J Kodjabachian. Incremental evolution of neural controllers for navigation in a 6-legged robot. In Sugisaka and Tanaka, editors, *Proc. of the Fourth International Symposium on Artificial Life and Robotics*. Oita Univ. Press, Japan, 1999. 70

Christopher D Fiorillo, William T Newsome, and Wolfram Schultz. The temporal precision of reward prediction in dopamine neurons. *Nature neuroscience*, 11 (8), July 2008. 179

D. Floreano and F. Mondada. Evolution of Plastic Neurocontrollers for Situated Agents. In P. Maes, M.J. Mataric, J.-A. Meyer, J. Pollack, and S.W. Wilson, editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB 96)*. MIT Press, 1996. 35, 109, 128

D. Floreano and J. Urzelai. Evolution and Learning in Autonomous Robots. In M. Tomassini and D. Mange, editors, *Bio-inspired Computing Machines*. PPUR, Lausanne, 1998. 35

D Floreano and J Urzelai. Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks : the official journal of the International Neural Network Society*, 13(4-5):431–43, 2000a. 75

D. Floreano and J. Urzelai. Evolution of Plastic Control Networks. *Autonomous Robots*, 11:311–317, 2001a. 75, 101

D. Floreano and J. Urzelai. Neural Morphogenesis, Synaptic Plasticity, and Evolution. *Theory in Biosciences*, 120:225–240, 2001b. 75

Dario Floreano and Laurent Keller. Evolution of adaptive behaviour in robots by means of Darwinian selection. *PLoS biology*, 8(1):e1000292, January 2010. 70, 75

Dario Floreano and Francesco Mondada. Automatic Creation of an Autonomous

Agent : Genetic Evolution of a Neural-Network Driven Robot. In Dave Cliff, editor, *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 421–430. MIT Press, 1994. 69, 75

Dario Floreano and Francesco Mondada. Evolutionary Neurocontrollers for Autonomous Mobile Robots. *Neural Networks*, 11(7-8):1461–1478, 1998. 75

Dario Floreano and Stefano Nolfi. Evolutionary Robotics. In Bruno Siciliano and Oussama Khatib, editors, *Handbook of Robotics*, chapter 61. Springer, 2008. 70, 75

Dario Floreano and Joseba Urzelai. Evolutionary robots: the next generation. *International Symposium on Evolutionary Robotics*, pages 1–26, 2000b. 70, 75

Dario Floreano, Francesco Mondada, Andres Perez-uribe, and Daniel Roggen. Machine Self-Evolution. *YLEM journal*, 24(12):4–10, 2004a. 70, 75

Dario Floreano, Francesco Mondada, Andres Perez-uribe, and Daniel Roggen. Evolution of Embodied Intelligence. In Fumiya Iida, Rolf Pfeifer, Luc Steels, and Yasuo Kuniyoshi, editors, *Embodied Artifcial Intelligence (Lecture Notes in Computer Science)*, pages 293–311. Springer-Verlag, Berlin, Heidelberg, 2004b. 70, 75

Dario Floreano, Jean-Christophe Zufferey, and Jean-Daniel Nicoud. From wheels to wings with evolutionary spiking circuits. *Artificial life*, 11(1-2):121–138, 2005. 75

Dario Floreano, Sara Mitri, Stéphane Magnenat, and Laurent Keller. Evolutionary conditions for the emergence of communication in robots. *Current biology : CB*, 17(6):514–9, March 2007. 29

Dario Floreano, Peter Dürr, and Claudio Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, January 2008. 70, 75

JA Fodor and Z W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988a. 43

Jerry Fodor and Brian P. McLaughlin. Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work. *Cognition*, 35(2):183–204, May 1990. 53

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, March 1988b. 53

David B. Fogel. Nils barricelli-artificial life, coevolution, self-adaptation. *Computational Intelligence Magazine, IEEE*, (February):41–43, 2006a. 29

David B. Fogel. Foundations of evolutionary computation. In *Proceedings of SPIE*, volume 6228, pages 622801–622801–13. SPIE, 2006b. 29, 32

David B. Fogel. Revisiting Overlooked Foundations of Evolutionary Computation: Part II. *Cybernetics and Systems*, 41(6):407–415, August 2010a. 33

David B. Fogel. Revisiting Overlooked Foundations of Evolutionary Computation: Part I. *Cybernetics and Systems*, 41(5):343–358, June 2010b. 33

David B. Fogel, Laurence J. Fogel, and V. W. Porto. Evolving Neural Networks. *Biological Cybernetics*, 63:487–493, 1990. 66

L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA, 1966. 31

Michael J Frank and Eric D Claus. Anatomy of a decision: striato-orbitofrontal interactions in reinforcement learning, decision making, and reversal. *Psychological review*, 113(2):300–26, April 2006. 181

S a Frank. The origin of synergistic symbiosis. *Journal of theoretical biology*, 176 (3):403–10, October 1995. 28

A. S. Fraser. Simulation of genetic systems by automatic digital computers : I Introduction. *Australian J. Biological Sciences*, 10(4):484–491, 1957a. 30

A. S. Fraser. Simulation of genetic systems by automatic digital computers : II. Effects of Linkage in Rates of Advance Under Selection. *Australian J. Biological Sciences*, 10(4):492–499, 1957b. 30

A. S. Fraser. Simulation of genetic systems by automatic digital computers : VII. Effects of Reporductive Rate, and Intensity of Selection on Genetic Structure. *Australian J. Biological Sciences*, 13(3):344–350, 1960a. 30

A. S. Fraser. Simulation of genetic systems by automatic digital computers : VI Epitstasis. *Australian J. Biological Sciences*, 13(2):344–350, 1960b. 30

Victor J Freeman. Studies on the virulence of bacteriophage-infected strains of corynebacterium diphtheriae. *Journal Of Bacteriology*, 61(6):675–688, 1951. 28

R M Friedberg. A Learning Machine: Part I. *IBM Journal of Research and Development*, 2(1):2–13, January 1958. 31

R M Friedberg, B Dunham, and J. H. North. A Learning Machine: Part II. *IBM Journal of Research and Development*, 3(3):282–287, July 1959. 31

Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews. Neuroscience*, 11(2):127–38, February 2010. 170, 214, 227

Karl Friston and Stefan Kiebel. Predictive coding under the free-energy principle. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1521):1211–21, May 2009. 213

Karl J Friston and Klaas E Stephan. Free-energy and the brain. *Synthese*, 159 (3):417–458, December 2007. 182, 210

Karl J Friston, Jean Daunizeau, and Stefan J Kiebel. Reinforcement learning or active inference? *PloS one*, 4(7):e6421, January 2009. 210

Tom Froese and Tom Ziemke. Enactive Artificial Intelligence. *Artifcial Intelligence*, 173(3-4):466–500, 2009. 57

J.C. Gallagher, R.D Beer, K.S. Espenschied, and R.D. Quinn. Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems*, 19:95–103, 1996a. 76

John C. Gallagher, Randall D. Beer, Kenneth S. Espenschied, and Roger D. Quinn. Autonomous Systems Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems*, 19:95–103, 1996b. 70

Peter Strick Garret E.Alexander, Mahlon R. DeLong. Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Neuroscience*, 9:357–381, 1986. 176, 306

M.I. Marta I MI Garrido, JM James M Kilner, SJ Stefan J S.J. Kiebel, KE Klaas E K.E. Stephan, Torsten Baldeweg, and K.J. Karl J Friston. Repetition suppression and plasticity in the human brain. *NeuroImage*, 48(1):269–279, October 2009. 220

David M. Gates. *Biophysical Ecology*. Dover Publications, London, 2003. 143, 219

J. Gibson. *The Theory of Affordances*, pages 67–82. 1977. 56

Onofrio Gigliotta, Giovanni Pezzulo, Stefano Nolfi, and Sefano Nolfi. Evolution of a predictive internal model in an embodied and situated agent. *Theory in biosciences = Theorie in den Biowissenschaften*, 130(4):259–76, December 2011. 74

J M Giménez-Amaya, N R McFarland, S de las Heras, and S N Haber. Organization of thalamic projections to the ventral striatum in the primate. *The Journal of comparative neurology*, 354(1):127–49, March 1995. 176, 301

Matthew R Ginther, Devin F Walsh, and Seth J Ramus. Hippocampal neurons encode different episodes in an overlapping sequence of odors task. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 31(7): 2706–11, February 2011. 339

D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, San Francisco, CA, 1987. Morgan Kaufmann. 81

Takashi Gomi and Ann Griffith. Evolutionary robotics-an overview. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 40–49. IEEE, 1996. 70

Giorgio Grasso and Michael Recce. Towards Genetically Evolved Dynamic Control for Quadruped Locomotion. *Connection Science*, 11(3-4):317–330, December 1999. 70

a M Graybiel. The basal ganglia and chunking of action repertoires. *Neurobiology of learning and memory*, 70(1-2):119–36, 1998. 213

Joachim De Greeff and Stefano Nolfi. Evolution of Communication in Robots. In P. Vargas, E. Di Paulo, I. Harvey, and P. Husbands, editors, *The Horizons for Evolutionary Robotics (in press)*, number ii. 2012. 29

K Grillspector, R Henson, and A Martin. Repetition and the brain: neural models of stimulus-specific effects. *Trends in Cognitive Sciences*, 10(1):14–23, January 2006. 220

S Grossberg. A neural network model of adaptively timed reinforcement learning and hippocampal dynamics. *CAS/CNS Technical Report Series*, 1992. 291

S Grossberg. The complementary brain: unifying brain dynamics and modularity. *Trends in cognitive sciences*, 4(6):233–246, June 2000. 340

S. Grossberg and N.A. Schmajuk. Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural Networks*, 2:79–102, April 1989. 186

Stephen Grossberg. *Predictions in the Brain: Using our past to generate a future*, chapter Cortical and Subcortical Predictive Dynamics and Learning during Perception, Cognition, Emotion and Action, pages 208–230. Oxford University Press, 2011. 291, 340

Moritz Grosse-Wentrup and Jose L Contreras-Vidal. The role of the striatum in adaptation learning: a computational model. *Biological cybernetics*, 96(4): 377–88, April 2007. 181

F. Gruau. Genetic synthesis of Boolean neural networks with a cell rewriting developmental process. *[Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pages 55–74, 1992. 66

F. Gruau. Automatic Definition of Modular Neural Networks. *Adaptive Behavior*, 3(2):151–183, September 1994. 36, 38, 70

Frederic Gruau and Kameel Quatramaran. Cellular encoding for interactive evolutionary robotics. In *Fourth European conference on artificial life*, pages 368–377. The MIT Press, 1997. 36

Aaron J Gruber, Sara a Solla, D James Surmeier, and James C Houk. Modulation of striatal single units by expected reward: a spiny neuron model displaying dopamine-induced bistability. *Journal of neurophysiology*, 90(2):1095–114, August 2003. 294, 295

Marc Guitart-Masip, Nico Bunzeck, Klaas E Stephan, Raymond J Dolan, and Emrah Düzel. Contextual novelty changes reward representations in the striatum. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 30(5):1721–6, February 2010. 179, 197

Suzanne N Haber and Roberta Calzavara. The cortico-basal ganglia integrative network: the role of the thalamus. *Brain research bulletin*, 78(2-3):69–74, February 2009. 176, 306

Heiko Hamann, J. Stradner, Thomas Schmickl, and Karl Crailsheim. A hormone-based controller for evolutionary multi-modular robotics: From single modules to gait learning. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010. 141

W D Hamilton. The genetical evolution of social behaviour. i. *Journal of Theoretical Biology*, 7(1):1–16, 1964. 25

P.J.B. Hancock. Genetic algorithms and permutation problems: a comparison of recombination operators for neural net structure specification. In *[Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks*, volume 1, pages 108–122. IEEE Comput. Soc. Press. 66

Bart L. M. Happel and Jacob M. J. Murre. Design and Evolution of Modular Neural Network Architectures. *Neural Networks.*, 7(6-7):985–1004, 1994. 36, 70

S A Harp, T Samad, and A Guha. *Towards the Genetic Synthesis of Neural*

Networks, pages 360–369. Number CSDD–89–I4852–2. Morgan Kaufmann, 1989. 66

Kyle I Harrington and Jordan B Pollack. Robot phylogenetics. In *Proceedings of the 12th annual conference comp on Genetic and evolutionary computation - GECCO '10*, volume 63, New York, New York, USA, 2010. ACM Press. 78

Kyle I Harrington, Lee Spector, Jordan B Pollack, and Una-may O Reilly. Autoconstructive Evolution for Structural Problems. In *Proceedings of the 13th annual conference comp on Genetic and evolutionary computation - GECCO '12*, Philadelphia, Pennsylvania, 2012. ACM Press. 78

L M Harrison, a Duggins, and K J Friston. Encoding uncertainty in the hippocampus. *Neural networks : the official journal of the International Neural Network Society*, 19(5):535–46, June 2006. 339

I. Harvey, P. Husbands, and D. Cliff. Issues in Evolutionary Robotics. In Jeanarcady Meyer, Herbert L. Roitblat, and Stewart W. Wilson, editors, *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Be- havior*, number July 1992, pages 364–373. A Bradford Book, 1993a. 69, 72

I. Harvey, P. Husbands, and D. Cliff. Genetic Convergence in a Species of Evolved Robot Control Architectures. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, page 636, San Francisco, CA, 1993b. Morgan Kaufmann Publishers Inc. 72

I. Harvey, P. Husbands, D. Cliff, and A. Thompson. Evolutionary robotics: the

Sussex approach. *Robotics and Autonomous Systems*, 20(June 1997):205–244, 1997. 70

Inman Harvey. Evolutionary Robotics and SAGA : the case for Hill Crawling and Tournament Selection (Cognitive Science Research Paper, 222). Technical report, University of Sussex, Brighton, 1992a. 69, 72

Inman Harvey. Species adaptation genetic algorithms: The basis for a continuing SAGA. In Francisco J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 346–354, 1992b. 72

Inman Harvey. *The artificial evolution of adaptive behaviour*. D.phil., University of Sussex, Brighton, 1993a. 73

Inman Harvey. *The Artificial Evolution of Adaptive Behaviour*. Dphil, University of Sussex, 1993b. 72

Inman Harvey. Is There Another New Factor in Evolution? *Evolutionary Computation*, 4(3):313–329, September 1996. 34, 73

Inman Harvey and Adrian Thompson. Through the Labyrinth Evolution Finds a Way : A Silicon Ridge. In T. Higuchi, M. Iwata, and L. Weixin, editors, *Proc. of The First International Conference on Evolvable Systems: From Biology to Hardware (ICES96).*, 1996. 36, 73

Inman Harvey, Eric Vaughan, and Ezequiel Di Paolo. Time and Motion Studies: The Dynamics of Cognition, Computation and Humanoid Walking. In *HART*

2004, Fourth Intl. Symp. on Human and Artificial Intelligence Systems: From Control to Autonomy, Fukui, Japan, 2004a. 73

Inman Harvey, Eric Vaughan, and Ezequiel Di Paolo. Time and motion studies: The dynamics of cognition, computation and humanoid walking. In *HART 2004, Fourth Intl. Symp. on Human and Artificial Intelligence Systems: From Control to Autonomy*, January 2004b. 73

Andrea Hasenstaub, Stephani Otte, Edward Callaway, and Terrence J Sejnowski. Metabolic cost as a unifying principle governing neuronal biophysics. *Proceedings of the National Academy of Sciences of the United States of America*, 107 (27):12329–34, July 2010. 220, 221

Erin J Hastings, Ratan K Guha, and Kenneth O Stanley. Automatic Content Generation in the Galactic Arms Race Video Game. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–19, 2009. 80

Sabine Hauert, Jean-Christophe Zufferey, and Dario Floreano. Evolved swarming without positioning information: anapplication in aerial communication relay. *Autonomous Robots*, 26(1):21–32, October 2008. 75

John Haugeland. *Artificial Intelligence: The Very Idea*. MIT Press, 1985. 50

Patrick Hayes. Trends in Artificial Intelligence. *Int. Journal Man-Machine Studies*, 10:295–299, 1976. 51

S Haykin. *Neural Networks: A Comprehensive Foundation*, volume 13. Prentice Hall, 1999. 38

Donald T. Haynie. *Biological Thermodynamics*. Cambridge University Press, Cambridge, Englsnd, 2001. 143, 219

Thomas E Hazy, Michael J Frank, and Randall C O'Reilly. Neural mechanisms of acquired phasic dopamine responses in learning. *Neuroscience and biobehavioral reviews*, 34(5):701–20, April 2010. 181, 291, 306, 307, 308

D O Hebb. Heredity and Environment in Mammalian Behaviour. *The British Journal of Animal Behaviour*, 1(2):43–47, 1953. 40

Donald O Hebb. *The Organization of Behavior: A Neuropsychological Theory*. John Wiley and Sons Inc., 1949. 40

M. Heidegger. *Being and time*. Wiley-Blackwell, 1962. 56

O Hikosaka. Basal ganglia–possible role in motor coordination and learning. *Current opinion in neurobiology*, 1(4):638–43, December 1991. 172

Jonathan D Hiller and Hod Lipson. Evolving Amorphous Robots. In Harold Fellermann, Mark Dörr, Martin M. Hanczyc, Lone Ladegaard Laursen, Sarah Maurer, Daniel Merkle, Pierre-Alain Monnard, Kasper Stoy, and Steen Rasmussen, editors, *Artificial Life XII Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 717–724. MIT Press, 2010. 78

Geoffrey E Hinton and Steven J Nowlan. How Learning Can Guide Evolution. *Complex Systems*, 1:495–502, 1987. 34

Julien Hirel, Philippe Gaussier, Mathias Quoy, and J.P. Banquet. Why and

how hippocampal transition cells can be used in reinforcement learning. *From Animals to Animats 11*, pages 359–369, 2010. 339

Masahiko Takada Hironobu Tokuno. Common structural organization of two output nuclei of primate basal ganglia. *Trends in Neurosciences*, 16(8):308–309, September 1993. 178

Meike Hoffmeister and William Martin. Interspecific evolution: microbial symbiosis, endosymbiosis and gene transfer. *Environmental microbiology*, 5:641–649, 2003. 27

T. Hoinville, C. Tapia Siles, and P. Henaff. Flexible and multistable pattern generation by evolving constrained plastic neurocontrollers. *Adaptive Behavior*, 19(3):187–207, May 2011. 141

David Holcman and Misha Tsodyks. The emergence of Up and Down states in cortical networks. *PLoS computational biology*, 2(3):e23, March 2006. 345

J H Holland. *Adaptation in Natural and Artificial Systems*, volume Ann Arbor. University of Michigan Press, 1975. 32

J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. NatL Acad. Sci. USA*, 79(April):2554–2558, 1982. 39, 43

Gregory S. Hornby and Jordan B. Pollack. Evolving L-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048, December 2001a. 78

Gregory S. Hornby and Jordan B. Pollack. Body-brain co-evolution using L-

systems as a generative encoding. *In Genetic and Evolutionary Computation Conference*, 2001b. 78

G.S. Hornby and J.B. Pollack. The advantages of generative grammatical encodings for physical design. *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, pages 600–607, 2001c. 78

AJ Horner. Priming, response learning and repetition suppression. *Neuropsychologia*, 46(7):1979–1991, 2008. 220

J C Houk, C Bastianen, D Fansler, A Fishbach, D Fraser, P J Reber, S A Roy, and L S Simo. Action selection and refinement in subcortical loops through basal ganglia and cerebellum. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 362(1485):1573–83, September 2007. 181

D.F. Hoyt and C.R. Taylor. Gait and the energetics of locomtion in horses. *Nature*, 292:239–240, 1981. 152

Sheng-Ta Hsieh, Tsung-Ying Sun, and Chan-Cheng Liu. Sharing evolution genetic algorithm for global numerical optimization. In *2008 IEEE Conference on Soft Computing in Industrial Applications*, number 1, pages 326–331. IEEE, June 2008. 81

Scott a Huettel, Allen W Song, and Gregory McCarthy. Decisions under uncertainty: probabilistic context influences activation of prefrontal and parietal cortices. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 25(13):3304–11, March 2005. 179

Mark D Humphries and Tony J Prescott. The ventral basal ganglia, a selection mechanism at the crossroads of space, strategy, and reward. *Progress in neurobiology*, 90(4):385–417, April 2010. 181

P. Husbands. Evolving Robot Behaviours with Diffusing Gas Networks. In P. Husbands and J.-A. Meyer, editors, *EvoRobot98: Proceedings of 1st European Workshop on Evolutionary Robotics*, pages 71–86. Springer-Verlag LNCS, 1998a. 116

P. Husbands, I. Harvey, and D. Cliff. Circle in the round: State space attractors for evolved sighted robots. *Robotics and Autonomous Systems*, 15(1-2):83–106, 1995. 72

P Husbands, I Harvey, D Cliff, and G Miller. Artificial evolution: a new path for artificial intelligence? *Brain and cognition*, 34(1):130–59, June 1997. 72

P. Husbands, T. Smith, N. Jakobi, and M. O'Shea. Better Living Through Chemistry: Evolving GasNets for Robot Control. *Connection Science*, 10(3-4):185–210, 1998a. 38

P. Husbands, T. Smith, M. O'Shea, M. Jakobi, J. Anderson, and A. Phillipedes. Brains Gases and Robots. In N. Niklasson et al, editor, *Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN98)*, pages 51–64. Springer-Verlag, 1998b. 119, 120, 128

P. Husbands, T.M.C.Smith, N. Jakobi, and M. O'Shea. Better Living Through Chemistry: Evolving GasNets for Robot Control. *Connection Science*, 10(3-4): 185–210, 1998c. 144

P. Husbands, A. Phillipedes, T. Smith, and M. O'Shea. Volume Signalling in Real and Robot Nervous Systems. *Theory in Biosciences*, 120:253–269, 2001a. 107

Phil Husbands. Evolving Robot Behaviours with Diffusing Gas Networks. In Philip Husbands and Jean-Arcady Meyer, editors, *Evolutionary Robotics: First European Workshop, EvoRobot 98, Proceedings (Lecture Notes in Computer Science)*, pages 71–86. Springer, 1998b. 72, 101

Phil Husbands, Tom Smith, Nick Jakobi, and Michael O'Shea. Better Living Through Chemistry: Evolving GasNets for Robot Control. *Connection Science*, 10(3-4):185–210, September 1998d. 72, 101

Phil Husbands, Tom Smith, Michael O Shea, Nick Jakobi, John Anderson, and Andy Philippides. Brains, Gases and Robots. In Lars Niklasson, Mikael Boden, and Tom Ziemke, editors, *ICANN'98:Proceedings of the 8th International Conference on Artificial Neural Networks, Skövde, Sweden, 2-4 September 1998 (Perspectives in Neural Computing)*, pages 51–63. Springer, 1998e. 37, 72, 99, 104

Phil Husbands, Andy Philippides, Tom Smith, and Michael O Shea. The Shifting Network: Volume Signalling in Real and Robot Nervous Systems. In Jozef Kelemen and Petr Sosik, editors, *Advances in Artificial Life: 6th European Conference, ECAL 2001, Proceedings*, pages 23–36. Springer, 2001b. 72, 104

Philip Husbands and I. Harvey. Evolution Versus Design: Controlling Autonomous Robots. In *Proceedings of the Third AnnualConference of AI, Simu-*

lation, and Planning in High Autonomy Syslems 'Integrating Perception, Planning and Action'., pages 139–146. IEEE Press, 1992. 69

Philip Husbands, Inman Harvey, Dave Cliff, and Geoffrey Miller. The Use of Genetic Algorithms for the Development of Sensorimotor Control Systems. In P. Gaussier and J. Nicoud, editors, *Proc. Int. Conf. on Perception to Action (PerAc'94)*, pages 110–121. IEEE Computer Society Press, 1994. 72

Philip Husbands, Owen Holland, and Michael Wheeler. Introduction: The Mechanical Mind. In Philip Husbands, Owen Holland, and Michael Wheeler, editors, *The mechanical mind in history*, chapter 1. MIT Press, Cambridge, MA, 2008. 45

Philip Husbands, Andrew Philippides, Patricia Vargas, C.L. Buckley, Peter Fine, E. Di Paolo, and M. O'Shea. Spatial, temporal, and modulatory factors affecting GasNet evolvability in a visually guided robotics task. *Complexity*, 16(2): 35–44, 2010. 72, 141

John Hutchins. ALPAC: the (in)famous report. In S. Nirenburg, H. Somers, and Y. Wilks, editors, *Readings in machine translation*, volume 14, pages 131–135. MIT Press, 2003. 50

Christian Igel. Neuroevolution for reinforcement learning using evolution strategies. *Evolutionary Computation, 2003. CEC'03. The 2003 ...*, 2003. 292

H. Iizuka and Ezekiel A Di Paolo. Toward Spinozist Robotics: Exploring the Minimal Dynamics of Behavioral Preference. *Adaptive Behavior*, 15(4):359–376, December 2007. 73

A.J. Ijspeert, John Hallam, and David Willshaw. Artificial lampreys: Comparing naturally and artificially evolved swimming controllers. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 256–265, 1997. 70

A. Ishiguro, A. Fujii, and P. EggenbergerHotz. Neuromodulated Control of Bipedal Locomotion Using a Polymorphic CPG Circuit. *Adaptive Behavior*, 11:7–17, 2003a. 102

Akio Ishiguro, Akinobu Fujii, and Peter Eggenberger Hotz. Neuromodulated Control of Bipedal Locomotion Using a Polymorphic CPG Circuit. *Adaptive Behavior*, 11(1):7–17, March 2003b. 38

Makoto Ito and Kenji Doya. Multiple representations and algorithms for reinforcement learning in the cortico-basal ganglia circuit. *Current opinion in neurobiology*, 21(3):368–373, April 2011. 181

Eugene M Izhikevich. Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral cortex (New York, N.Y. : 1991)*, 17(10): 2443–52, October 2007. 240

E. Izquierdo, I. Harvey, and R. D. Beer. Associative Learning on a Continuum in Evolved Dynamical Neural Networks. *Adaptive Behavior*, 16(6):361–384, December 2008. 77, 292

Eduardo Izquierdo. The dynamics of associative learning in an evolved situated agent. *Advances in Artificial Life*, (1), 2007. 292

Eduardo Izquierdo-Torres and Ezequiel Di Paolo. Is an Embodied System Ever Purely Reactive? In Mathieu Capcarrère, Alex Freitas, Peter Bentley, Colin

Johnson, and Jon Timmis, editors, *Advances in Artifcial Life (Lecture Notes in Computer Science)*, volume 3630, pages 252–261. Springer, Berlin / Heidelberg, 2005. 73

Nick Jakobi, Phil Husbands, and Tom Smith. Robot Space Exploration by Trial and Error. In John R. Koza, Forrest H Bennett, and David Andre, editors, *Proceedings of the Third Annual Conference on Genetic Programming: GP98*, pages 807–815. Morgan Kaufmann, 1998. 72

Derek James and Philip Tucker. A Comparative Analysis of Simplification and Complexification in the Evolution of Neural Network Topologies. In Kalyanmoy Deb, Riccardo Poli, Wolfgang Banzhaf, Hans-Georg Beyer, Edmund Burke, Paul Darwen, Dipankar Dasgupta, Dario Floreano, and James Foster, editors, *Genetic and Evolutionary Computation - GECCO 2004: Genetic and Evolutionary Computation Conference*, Seattle, Washington, USA, 2004. Springer. 80

T Jay. Dopamine: a potential substrate for synaptic plasticity and memory mechanisms. *Progress in Neurobiology*, 69(6):375–390, April 2003. 179

J. Jing, F.S. Vilim, E.C. Cropper, and K.R. Weiss. Neural analog of arousal: persistent conditional activation of a feeding modulator by serotonergic initiators of locomotion. *The Journal of Neuroscience*, 28(47):12349–12361, 2008. 197

Michael I. Jordan. Serial order: A parallel distributed processing approach. Technical report, California University San Diego, La Jolla Institute for Cognitive Science, 1986. 290

S.E. Jorgensen and G. Bendoricchio. *Fundamentals of Ecological Modelling, Third Edition (Developments in Environmental Modelling*. Elsevier Science, Oxford, England, 2001. 143, 219

D. Joseph, G. Gangadhar, and V. Srinivasa Chakravarthy. ACE (Actor-Critic-Explorer) paradigm for reinforcement learning in basal ganglia: Highlighting the role of subthalamic and pallidal nuclei. *Neurocomputing*, 74:205–218, 2010. 181

Roberto R Kampfner and Michael Conrad. Computational Model of Evolutionary Learning Processes in the Brain. *Bulletin of Mathematical Biology*, 45(6):931–968, 1983. 67

Fred A Keijzer. Some Armchair Worries about Wheeled Behavior. *Proceedings of the fifth international conference on simulation of adaptive behavior on (From animals to animats 5)*, pages 13–21, 1998. 61

Steven W Kennerley and Jonathan D Wallis. Reward-dependent modulation of working memory in lateral prefrontal cortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 29(10):3259–70, March 2009. 179

O. Kiehn and P. Katz. Making circuits dance: the neuromodulation of motor systems. In P. Katz, editor, *Beyond Neurotransmission*, pages 275–313. Oxford University Press, 1999. 152

R Kim, K Nakano, a Jayaraman, and M B Carpenter. Projections of the globus pallidus and adjacent structures: an autoradiographic study in the monkey. *The Journal of comparative neurology*, 169(3):263–90, October 1976. 176, 301

M. Kimura. *"The Neutral Theory of Molecular Evolution"*. "Cambridge University Press", "New York", 1983. 35

Kevin G Kirby and Michael Conrad. Intraneuronal dynamics as a substrate for evolutionary learning. *Physica D: Nonlinear Phenomena*, pages 205–215, 1986. 66

Hiroaki Kitano. Designing Neural Networks Using Genetic Algorithms with Graph Generation System. *Complex Systems*, 4:461–476, 1990. 66

Philip Kitcher, R Singh, C Krimbas, Diane B Paul, and John Beatty. *Battling the Undead. How (And How Not) to Resist Genetic Determinism*, pages 396–414. Cambridge University Press, 2001. 27

Ronald Kline. Cybernetics, Automata Studies, and the Dartmouth Conference on Artificial Intelligence. *IEEE Annals of the History of Computing*, 33(4): 5–16, April 2011. 45

T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990. 39, 43

Teuvo Kohonen. Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43(1):59–69, 1982. 43

John R Koza. Hierarhical Genetic Algorithms Operating on Populations of Computer Programs. *IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence*, 1, 1989. 32

John R Koza and Riccardo Poli. A Genetic Programming Tutorial. In Edmond

Burke, editor, *Introductory Tutorials in Optimization Search and Decision Support*, pages 55—-95. Kluwer Academic Press, 2003. 32

JR Koza. Genetically breeding populations of computer programs to solve problems in artificial intelligence. Technical Report 1, Technical Report, Stanford University, Stanford, 1990. 32

Gabriel Kreiman. Neural coding: computational and biophysical perspectives. *Physics of Life Reviews*, 1(2):71–102, July 2004. 220

T.S. Kuhn. *The structure of scientific revolutions*. University of Chicago press, 1996. 55

Dharshan Kumaran and Emrah Duzel. The hippocampus and dopaminergic midbrain: old couple, new insights. *Neuron*, 60(2):197–200, October 2008. 180

Ibrahim Kuscu and Chris Thornton. Design of artificial neural networks using genetic algorithms: Review and prospect. *Neural Networks*, 1994. 37, 66

C J Lacey, J Boyes, O Gerlach, L Chen, P J Magill, and J P Bolam. GABA(B) receptors at glutamatergic synapses in the rat striatum. *Neuroscience*, 136(4): 1083–95, January 2005. 300

José L Lanciego, Nancy Gonzalo, Maria Castle, Carlos Sanchez-Escobar, Maria S Aymerich, and Jose a Obeso. Thalamic innervation of striatal and subthalamic neurons projecting to the rat entopeduncular nucleus. *The European journal of neuroscience*, 19(5):1267–77, March 2004. 176, 301

Russell Lande. Adaptation to an extraordinary environment by evolution of

phenotypic plasticity and genetic assimilation. *Journal of evolutionary biology*, 22(7):1435–46, July 2009. 34

S B Laughlin. Energy as a constraint on the coding and processing of sensory information. *Current opinion in neurobiology*, 11(4):475–80, August 2001. 221

S.B. Laughlin, J.C. Anderson, D O'Carroll, and R De Ruyter Van Steveninck. *Information Theory and the Brain*, chapter Coding Ffficiency and the Metabolic Cost of Sensory and Neural Information. Cambridge University Press, 2000. 221

Simon B Laughlin and Terrence J Sejnowski. Communication in neuronal networks. *Science (New York, N.Y.)*, 301(5641):1870–4, September 2003. 219, 221

Joel Lehman and Kenneth O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11*, (Gecco): 211, 2011a. 38

Joel Lehman and Kenneth O Stanley. Abandoning objectives: evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, January 2011b. 83

E G Leigh. Neutral theory: a historical perspective. *Journal of evolutionary biology*, 20(6):2075–91, November 2007. 36

Peter Lennie. The cost of cortical computation. *Current biology*, 13(6):493–497, 2003. 220

Jose León-Carrión, Meltem Izzetoglu, Kurtulus Izzetoglu, Juan Francisco Martín-Rodríguez, Jesús Damas-López, Juan Manuel Barroso y Martin, and María Rosario Domínguez-Morales. Efficient learning produces spontaneous neural repetition suppression in prefrontal cortex. *Behavioural brain research*, 208(2):502–8, April 2010. 220

W B Levy and R a Baxter. Energy efficient neural codes. *Neural computation*, 8 (3):531–43, April 1996. 220

B L Lewis and P O'Donnell. Ventral tegmental area afferents to the prefrontal cortex maintain membrane potential 'up' states in pyramidal neurons via D(1) dopamine receptors. *Cerebral cortex (New York, N.Y. : 1991)*, 10(12):1168–75, December 2000. 345

M Anthony Lewis, Andrew H Fagg, and George A Bekey. Genetic Algorithms for Gait Synthesis in a Hexapod Robot. In Zheng, editor, *Recent Trends in Mobile Robots*, pages 317–331. World Scientific, New Jersey, 1994. 69

R C Lewontin. Directions in evolutionary biology. *Annual review of genetics*, 36: 1–18, January 2002. 28

R C Lewontin and L C Dunn. The evolutionary dynamics of a polymorphism in the house mouse. *Genetics*, 45(6):705–722, 1960. 30

J Lighthill. Artificial intelligence: A general survey. *Artificial Intelligence a paper symposium*, page 121, 1973. 51

H Lipson. Principles of modularity, regularity, and hierarchy for scalable systems. *Journal of Biological Physics and Chemistry*, 7(4):125–128, June 2007. 78

H. Lipson, J.B. Pollack, and N. P Suh. Promoting Modularity in Evolutionary Design. *Proceedings of DETC01 2001 ASME Design Engineering Technical Conferences*, 2001. 78

Hod Lipson and Jordan B Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(August):974–978, 2000. 78

Hod Lipson, Jordan B. Pollack, and Nam P. Suh. On the origin of modular variation. *Evolution, International Journal of Organic Evolution*, 56(8):1549–1556, 2002. 78

Hod Lipson, Josh Bongard, Victor Zykov, and Evan Malone. Evolutionary Robotics for Legged Machines : From Simulation to Physical Reality. In *Proceedings of the 9th Int. Conference on Intelligent Autonomous Systems*, pages 11–18, University of Tokyo, Tokyo, Japan, 2006. 78

John Lisman and a D Redish. Prediction, sequences and the hippocampus. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1521):1193–201, May 2009. 339

Robert Lowe, Alberto Montebelli, Ioannis Ieropoulos, John Greenman, Chris Melhuish, and Tom Ziemke. Grounding Motivation in Energy Autonomy: A Study of Artificial Metabolism Constrained Robot Dynamics. *Fellermann, H.; Dörr, M.; Hanczyc, M.; Laursen, LL*, pages 725–732, 2010. 141

ChristopherJ. MacDonald, KyleQ. Lepage, UriT. Eden, and Howard Eichenbaum. Hippocampal Time Cells Bridge the Gap in Memory for Discontiguous Events. *Neuron*, 71(4):737–749, August 2011. 339

Brenda Maddox. The double helix and the wronged heroine. *Nature*, 421(6921): 407–408, 2003. 25

Sven Magg and Andrew Philippides. GasNets and CTRNNs A Comparison in Terms of Evolvability. *From Animals to Animats 9*, pages 1–12, 2006. 141

Karen Magnuson-Ford, Arne Mooers, Sébastien Rioux Paquette, and Mike Steel. Comparing strategies to preserve evolutionary diversity. *Journal of theoretical biology*, 266(1):107–16, September 2010. 37

Edward O Mann and Ole Paulsen. Role of GABAergic inhibition in hippocampal network oscillations. *Trends in neurosciences*, 30(7):343–9, July 2007. 222

L Margulis. Genetic and evolutionary consequences of symbiosis. *Experimental parasitology*, 349:277–349, 1976. 27

Lynn Margulis. *Gaia is a tough bitch*, page 413. Touchstone, 1995. 27

H. Markham and M. Tsodyks. Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature*, 382:807–810, 1996. 240

B. Mathayomchan and R.D. Beer. Center-Crossing Recurrent Neural Networks for the Evolutionof Rhythmic Behavior. *Neural Computation*, 14:2043–2051, 2002a. 106, 117

Boonyanit Mathayomchan and Randall D. Beer. Center-Crossing Recurrent Neural Networks for the Evolution of Rhythmic Behavior. *Neural Computation*, 14 (9):2043–2051, September 2002b. 76, 99, 101

K. Matsuoka. Sustained-Oscillations Generated by Mutually Inhibiting Neurons with Adaptation. *Biologocal Cybernetics*, 52:367–376, 1985. 112

H Maturana. Autopoiesis, structural coupling and cognition: a history of these and other notions in the biology of cognition. *Cybernetics & Human Knowing*, 9(3):5–34, 2002. 62

Humberto R Maturana. Biology of Cognition. Technical Report 1970, Biological Computer Laboratory Research Report BCL 9.0, Urbana IL: University of Illinois, 1970. 60, 61

Humberto R Maturana and Francisco J Varela. *The Tree of Knowledge*, volume 9. Shambhala, 1987. 60

Giles Mayley. Landscapes, Learning Costs, and Genetic Assimilation. *Evolutionary Computation*, 4(3):213–234, September 1996. 34, 73

J Maynard Smith. Evolution and theory of games. *American Scientist*, 64:41–45, 1982. 26

J Maynard-Smith. The genetics of stasis and punctuation. *Annual review of genetics*, 17:11–25, 1983. 36

John Maynard Smith. Natural Selection and the Concept of a Protein Space. *Nature*, 225(5232):563–564, February 1970. 35

J. McCarthy, M. L. Minsky, N. Rochester, and C.E. Shannon. A proposal for the Dartmouth summer research project on artifcial intelligence, 1955. 49

Michael McCloskey. NETWORKS AND THEORIES: The Place of Connectionism to Cognitive Science. *Psychological Science*, 2(6):387–395, November 1991. 39

Warren S Mcculloch. Recollections of the Many Sources of Cybernetics. *Russell The Journal Of The Bertrand Russell Archives*, VI(2):5–16, 1974. 23, 41, 45

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, December 1943. 41, 46

WS McCulloch. Biological computers. *Electronic Computers, IRE Transactions on Electronic Computers*, pages 190–192, 1957. 41

John G McHaffie, Terrence R Stanford, Barry E Stein, Véronique Coizet, and Peter Redgrave. Subcortical loops through the basal ganglia. *Trends in neurosciences*, 28(8):401–7, August 2005. 176

G. McHale and P. Husbands. Quadrupedal Locomotion: GasNets, CTRNNs and Hybrid CTRNN/PNNs Compared. In P. Husbands T. Ikegami J. Pollack, M.A. Bedau and R.A. Watson, editors, *Artificial Life IX : Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 106–116. MIT Press, 2004a. 111

G. McHale and Phil Husbands. Incorporating energy expenditure into evolutionary robotics fitness measures. In *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 206–212, 2006. iii

G.A. McHale. A comparison of evolvable neural networks applied to bipedal locomation. Master's thesis, Evolutionary and Adptive Systems, COGS, University of Sussex, Brghton, UK, 2000. 112

Gary McHale and Phil Husbands. Quadrupedal locomotion: GasNets, CTRNNs and hybrid CTRNN/PNNs compared. *Proceedings of the 9th International Conference on Artifcial Life*, pages 106–112, 2004b. iii, 35, 38, 73, 141

Gary McHale and Phil Husbands. GasNets and other Evolvable Neural Networks applied to Bipedal Locomotion. In Stefan Schaal, editor, *Animals to animats 8:Proceedings of the 8th international conference on the simulation of adaptive behaviour (SAB2004).*, pages 163–172, Cambridge, 2004c. MIT Press. iii, 35, 38, 73, 141

David A Medler. A Brief History of Connectionism. *Neural Computing Surveys*, 1(2):18–72, 1998. 39

M Melissa Flesher, Allen E Butt, and Brandee L Kinney-Hurd. Differential acetylcholine release in the prefrontal cortex and hippocampus during pavlovian trace and delay conditioning. *Neurobiology of learning and memory*, 96(2):181–91, September 2011. 339

G Mendel. Experiments in plant hybridization (1865). *Read at the meetings of February 8th and March 8th*, (1865):3–47, 1865. 25

E Mengual, S de las Heras, E Erro, J L Lanciego, and J M Giménez-Amaya. Thalamic interaction between the input and the output systems of the basal ganglia. *Journal of chemical neuroanatomy*, 16(3):187–200, May 1999a. 176

E Mengual, S de las Heras, E Erro, J L Lanciego, J M Giménez-Amaya, and E Scarnati. The thalamus as a place for interaction between the input and the output systems of the basal ganglia: a commentary. *Journal of chemical neuroanatomy*, 16(3):187–200, May 1999b. 301

A Michael and Michael A Arbib. Warren McCullochs Search for the Logic of the Nervous System. *Perspectives in biology and medicine*, 43(2):193–216, 2000. 41

Thomas Miconi. *The Road to Everywhere: Evolution, Complexity and Progress in Natural and Artificial Systems*. Ph.d. thesis, University of Birmingham, 2007. 70

Orazio Miglino, Stefano Nolfi, and Domenico Parisi. Discontinuity in evolution : how different levels of organization imply pre-adaptation. In R.K. Belew and M. Mitchell, editors, *Adaptive Individuals in Evolving Populations*, pages 399–415. Addison-Wesley, Readings, MA, 1996. 73

M Migliore and M Culotta. Energy efficient modulation of dendritic processing functions. *Bio Systems*, 48(1-3):157–63, 1998. 220

B.L. Miller and M.J. Shaw. Genetic algorithms with dynamic niche sharing for multimodal function optimization. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 786–791. IEEE, 1996. 81

M L Minsky and S A Papert. *Perceptrons*. The MIT Press, 1969. 42

Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49 (1):8–30, 1961. 49

J. Mirenowicz and W. Schultz. Importance of unpredictability for reward responses in primate dopamine neurons. *Journal of Neurophysology*, 72(2), August 1994. 240

Sara Mitri, Dario Floreano, and Laurent Keller. Relatedness influences signal

reliability in evolving robots. *Proceedings. Biological sciences / The Royal Society*, 278(1704):378–83, February 2011. 75

R.C. Moioli, P.A. Vargas, and Phil Husbands. Exploring the Kuramoto model of coupled oscillators in minimally cognitive evolutionary robotics tasks. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010. 73, 141

Renan C. Moioli, Patricia a. Vargas, Fernando J. Von Zuben, and Phil Husbands. Towards the evolution of an artificial homeostatic system. *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 4023–4030, June 2008a. 73, 141

Renan C Moioli, Patricia A Vargas, and Fernando J Von Zuben. Evolving an Artificial Homeostatic System. In Gerson Zaverucha and Augusto Loureiro Costa, editors, *SBIA '08 Proceedings of the 19th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, pages 278–288, Berlin, Heidelberg, 2008b. Springer-Verlag. 141

Renan C. Moioli, Patricia a. Vargas, and Phil Husbands. A multiple hormone approach to the homeostatic control of conflicting behaviours in an autonomous mobile robot. *2009 IEEE Congress on Evolutionary Computation*, pages 47–54, May 2009. 73

Francesco Mondada and Dario Floreano. Evolution of neural control structures: some experiments on mobile robots. *Robotics and Autonomous Systems*, 1995. 69, 75

Francesco Mondada and Edoardo Franzi. Biologically Inspired Mobile Robot Control Algorithms. In *Proceddings of the NRP23 - Symposium on Artifcial Intelligence and Robotics*, Switzerland, 1993. 69

David J Montana and Lawrence Davis. Training Feedforward Neural Networks Using Genetic Algorithms. *IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence*, pages 762–767, 1989. 66

GE Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 38(8):114–117, 1965. 51

R.Y. Moore and F.E. Bloom. Central catecholamine neuron systems: anatomy and physiology of the norepinephrine and epinephrine systems. *Annual review of neuroscience*, 1979. 171

Roberto Moreno-Díaz and Arminda Moreno-Díaz. On the legacy of W.S. McCulloch. *Bio Systems*, 88(3):185–90, April 2007. 46

Lindsay K Morgan, Sean P Macevoy, Geoffrey K Aguirre, and Russell a Epstein. Distances between real-world locations are represented in the human hippocampus. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 31(4):1238–45, January 2011. 339

Jonathan Moss and J Paul Bolam. A dopaminergic axon lattice in the striatum and its relationship with cortical and thalamic terminals. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 28(44): 11221–30, October 2008. 300

Ulrich Mueller and K H. Grobman. Artificial life and Piaget. *Trends in cognitive sciences*, 7(4):149–151, April 2003. 74

Efstathios Mytilinaios, Mark Desnoyer, David Marcus, and Hod Lipson. Designed and Evolved Blueprints For Physical Self-Replicating Machines. In Jordan Pollack, Mark A. Bedau, Phil Husbands, Takashi Ikegami, and Richard A. Watson, editors, *Artificial Life IX, Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 15–20. MIT Press, 2004. 78

Atsushi Nambu, Hironobu Tokuno, and Masahiko Takada. Functional significance of the cortico-subthalamo-pallidal 'hyperdirect' pathway. *Neuroscience research*, 43(2):111–7, June 2002. xvi, xix, 173, 177

Nandakumar S Narayanan, Douglas J Guarnieri, and Ralph J DiLeone. Metabolic hormones, dopamine circuits, and feeding. *Frontiers in neuroendocrinology*, 31 (1):104–12, January 2010. 180

Natalia Caporale And Yang Dan. Spike TimingDependent Plasticity: A Hebbian Learning Rule. *Annual review of neuroscience*, 31:25–46, 2008. 238

W.J.H. Nauta and W.R Mehler. Projections of the lentiform nucleus in the monkey. *Brain Research*, Volume 1(Issue 1):3–42, 1996. 176

Andrew L. Nelson, Gregory J. Barlow, and Lefteris Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370, April 2009. 70

A Newell and H A Simon. GPS, a program that simulates human thought. In

Edward A. Feigenbaum and Julian Feldman, editors, *Computers and Thought*, pages 279–293. McGraw-Hill, 1963. 50

A Newell, J C Shaw, and H A Simon. Report on a general problem-solving program. *Proceedings of the International Conference on Information Processing*, 2(5222):256–264, 1959. 49

Andre Nieoullon. Dopamine and the regulation of cognition and attention. *Progress in neurobiology*, 571:1–31, November 2002. 179

G. Nitschke. Co-evolution of cooperation in a pursuit evasion game. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 2:2037–2042, 2003. 29

Y. Niv, D. Joel, I. Meilijson, and E. Ruppin. Evolution of Reinforcement Learning in Uncertain Environments: A Simple Explanation for Complex Foraging Behaviors. *Adaptive Behavior*, 10(1):5–24, January 2002. 292

Yael Niv. Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154, June 2009. 181

Jeremy E Niven and Simon B Laughlin. Energy limitation as a selective pressure on the evolution of sensory systems. *The Journal of experimental biology*, 211 (Pt 11):1792–804, June 2008. 220

S Nolfi and D Floreano. Coevolving predator and prey robots: do "arms races" arise in artificial evolution? *Artificial life*, 4(4):311–35, January 1998. 75

S Nolfi and D Parisi. *Artifcial Life III*, chapter Growing Neural Networks. Santa Fe Institute Series. Westview Press, 1994. 69

S Nolfi, G Baldassarre, and D Marocco. The Importance of Viewing Cognition as the Result of Emergent Processes Occurring at Different Time Scales. *Connection Science*, 14(3):231–244, 2002. 74

Stefano Nolfi. Learning and Evolution in Neural Networks. *Adaptive Behavior*, 3 (1):5–28, 1997. 34

Stefano Nolfi and Dario Floreano. Synthesis of autonomous robots through evolution. *Trends in Cognitive Sciences*, 6(1):31–37, 2002. 70, 75

Stefano Nolfi and Domenico Parisi. Growing neural networks (Technical Report PCIA-91-15). Technical report, National Research Council, Department of Cognitive Processes and Artificial Intelligence, Rome, 1991. 69

Stefano Nolfi and Domenico Parisi. Evolving non-trivial behaviors on real robots: an autonomous robot that picks up objects. In *Topics in Artificial Intelligence: Fourth Congress of the Italian Association for Artificial Intelligence*, number 1992, pages 243–255. 1995a. 73

Stefano Nolfi and Domenico Parisi. 'Genotypes' for neural networks. In M. A. Arbib, editor, *The handbook of brain theory and neural networks*, pages 1–14. MIT Press, Cambridge, MA, 1st editio edition, 1995b. 66

Stefano Nolfi and Domenico Parisi. Evolving Artificial Neural Networks that Develop in Time. In *Proceedings of the Third European Conference on Artificial Life*, pages 353–367, Berlin, 1995c. Springer-Verlag. 73

Stefano Nolfi and Domenico Parisi. Learning to adapt to changing environments in evolving neural networks (Technical Report 95-15). Technical report, De-

partment of Neural Systems and Artificial Life, National Research Council, Rome, 1995d. 73

Stefano Nolfi and Domenico Parisi. Neural Networks in an Artificial Life Perspective. In W. Gerstner, A. Germond, M. Hasler, and J.D. Nicoud, editors, *Artificial Neural Networks (ICANN97). Proceedings of the 7th International Conference on Artificial Neural Networks*, number c, pages 733–738, Berlin, Heidelberg, 1997a. Springer-Verlag. 73

Stefano Nolfi and Domenico Parisi. Evolution of Artificial Neural Networks. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 418–421. MIT Press, Cambridge, MA, 2 edition, 1997b. 73

Stefano Nolfi, Jeffrey L Elman, and Domenico Parisi. Learning and evolution in neural networks. *Adaptive Behavior*, 3(1):5–28, 1994a. 73

Stefano Nolfi, Orazio Miglino, and Domenico Parisi. Phenotypic Plasticity in Evolving Neural Networks. In *Proceedings of the International Conference From Perception to Action.*, pages 146–157, Los Alamitos, CA, 1994b. IEEE Computer Society Press. 73

JP Nordin and M. Nordahl. An Evolutionary Architecture for a Humanoid Robot. In *Proceeding of: The Fourth International Symposium on Artificial Life and Robotics (AROB 4th 99) Oita Japan*, 1999. 70

Patricio O'Donnell. Dopamine gating of forebrain neural ensembles. *European Journal of Neuroscience*, 17(3):429–435, February 2003. 345

Murat Okatan. Correlates of reward-predictive value in learning-related hippocampal neural activity. *Hippocampus*, 19(5):487–506, May 2009. 339

Randall C O'Reilly and Michael J Frank. Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural computation*, 18(2):283–328, February 2006. 181

G Palm. Evidence, Information, and Surprise. *Biological Cybernetics*, 68:57–68, 1981. 227

Richard D Palmiter. Is dopamine a physiologically relevant mediator of feeding behavior? *Trends in neurosciences*, 30(8):375–81, August 2007. 180

Z Pang, G Y Ling, M Gajendiran, and Z C Xu. Enhanced excitatory synaptic transmission in spiny neurons of rat striatum after unilateral dopamine denervation. *Neuroscience letters*, 308(3):201–5, August 2001. 295

S Papert. One AI or many? *Daedalus (Boston)*, 117(1):1–14, 1988. 42, 43

a Parent and L N Hazrati. Anatomical aspects of information processing in primate basal ganglia. *Trends in neurosciences*, 16(3):111–6, March 1993. 178

Domenico Parisi. Simulating the evolutionary emergence of language: a research agenda. In A. Cangelosi, A.D.M. Smith, and K. Smith, editors, *Evolution of Language*, pages 230–238, Singapore, 2006. World Scientific. 74

Domenico Parisi and Stefano Nolfi. The influence of learning on evolution (Technical Report PCIA-94-09). Technical report, Department of Cognitive Processes and Artificial Intelligence, National Research Council, Rome, 1994. 73

Domenico Parisi, Stefano Nolfi, and Federico Cecconi. Learning, Behavior, and Evolution. In F Valera and P Bourgine, editors, *Toward a practice of autonomous systems proceedings of the first European Conference on Artificial Life*, pages 207–217. MIT Press, 1992. 68, 73

Judea Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, 2 edition, 2009. 312

C.M.a. Pennartz, R. Ito, P.F.M.J. Verschure, F.P. Battaglia, and T.W. Robbins. The hippocampalstriatal axis in learning, prediction and goal-directed behavior. *Trends in Neurosciences*, 34(10):548–559, September 2011. 339

A. Petrowski. A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE International Conference on Evolutionary Computation ICEC-96*, pages 798–803. IEEE, 1996. 81

Rolf Pfeifer. Building "fungus" eaters: Design principles of autonomous agents. In Pattie Maes, Maja J. Mataric, Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson, editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior SAB96*, volume 4, pages 3–12, 1996. 58, 69

P. Phattanasri, H. J. Chiel, and R. D. Beer. The Dynamics of Associative Learning in Evolved Model Circuits. *Adaptive Behavior*, 15(4):377–396, December 2007a. 342

P. Phattanasri, H. J. Chiel, and R. D. Beer. The Dynamics of Associative Learning in Evolved Model Circuits. *Adaptive Behavior*, 15(4):377–396, December 2007b. 77

A. Philippedes, P. Husbands, T. Smith, and M. O'Shea. Fast and Loose : Biologically Inpsired Couplings. In R. Standish, M.A. Bedau, and H. A. Abbass, editors, *Artificial Life VIII: Proceedings of the Eight International Conference on Artificial Life*, pages 292–301. MIT Press, 2002. 136, 139

Andy Philippides, Phil Husbands, Tom Smith, and Michael O Shea. Fast and Loose : Biologically Inspired Couplings. In Russell Standish, Hussein A. Abbass, and Mark A. Bedau, editors, *Artificial Life VIII: Proceedings of the Eighth International Conference on Artificial Life (Complex Adaptive Systems)*, pages 293–301. A Bradford Book, 2002. 72, 104

Andy Philippides, Phil Husbands, Tom Smith, and Michael O'Shea. Flexible couplings: diffusing neuromodulators and adaptive robotics. *Artificial life*, 11 (1-2):139–60, 2005. 72, 141

Gualtiero Piccinini. THE FIRST COMPUTATIONAL THEORY OF MIND AND BRAIN : A CLOSE LOOK AT MCCULLOCH AND PITTSS LOGICAL CALCULUS OF IDEAS IMMANENT IN NERVOUS ACTIVITY. *Synthese*, (i):175–215, 2004. 46

Massimo Pigliucci. Do we need an extended evolutionary synthesis? *Evolution; international journal of organic evolution*, 61(12):2743–9, December 2007. 28

Alex Pine, Ben Seymour, Jonathan P Roiser, Peter Bossaerts, Karl J Friston, H Valerie Curran, and Raymond J Dolan. Encoding of marginal utility across time in the human brain. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 29(30):9575–81, July 2009. 180

Alex Pine, Tamara Shiner, Ben Seymour, and Raymond J Dolan. Dopamine, time, and impulsivity in humans. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 30(26):8888–96, June 2010. 179

Jordan B Pollack. Connectionism: Past, Present, and Future. *Artificial Intelligence Review*, 3(1):3–20, 1988. 77

Jordan B. Pollack, Hod Lipson, Gregory Hornby, and Pablo Funes. Three Generations of Automatically Designed Robots. *Artificial Life*, 7(3):215–223, July 2001. 78

Thomas E Portegys. A maze learning comparison of Elman, long short-term memory, and Mona neural networks. *Neural networks: the official journal of the International Neural Network Society*, 23(2):306–13, March 2010. 290

DK Pratihar. Evolutionary robotics A review. *Sadhana*, 28(December):999–1009, 2003. 70

Sean Psujek and Randall D. Beer. Developmental bias in evolution: evolutionary accessibility of phenotypes in a model evo-devo system. *Evolution & Development*, 10(3):375–390, May 2008. 76

Sean Psujek, Jeffrey Ames, and Randall D. Beer. Connection and Coordination: The Interplay Between Architecture and Dynamics in Evolved Model Pattern Generators. *Neural Computation*, 18(3):729–747, March 2006. 76

M. Quinn, L. Smith, G. Mayley, and P. Husbands. Evolving Teamwork and Role Allocation with Real Robots (Cognitive Science Research Paper 515). page 10,

Brighton. School of Cognitive and Computing Sciences, University of Sussex. 73

Matt Quinn, Lincoln Smith, Giles Mayley, and Phil Husbands. Evolving controllers for a homogeneous system of physical robots: structured cooperation with minimal sensors. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 361(1811):2321–43, October 2003. 73

Marcus E Raichle. The Brains Dark Energy. *Science*, 1249(2006), 2006. 228, 345

Marcus E Raichle and Debra A Gusnard. Appraising the brains energy budget. *PNAS*, 99(16):10237–10239, 2002. 228

Marcus E Raichle and Abraham Z Snyder. A default mode of brain function: a brief history of an evolving idea. *NeuroImage*, 37(4):1083–90; discussion 1097–9, October 2007. 221

Jaime David Gomez Ramrez. *A New Foundation for Representation in Cognitive and Brain Science: Category Theory and the Hippocampus*. PhD thesis, Universidad Politecnica de Madrid, 2010. 314

N. Rashevsky. Outline of a physico-mathematical theory of excitation and inhibition. *Protoplasma*, 20(1):42–55, 1933. 40

N. Rashevsky. Outline of a Physico-Mathematical Theory of the Brain. *Journal of General Psychology*, 13:82–112, 1935. 40

I Rechenberg. Cybernetic solution path of an experimental problem. (royal aircraft establishment translation no. 1122, bf toms, trans.). *Farnsborough Hants Ministery of Aviation Royal Aircraft Establishment*, 1122, 1965. 31

I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD thesis, Technical University of Berlin, 1971. 31

P Redgrave, T J Prescott, and K Gurney. Is the short-latency dopamine response too short to signal reward error? *Trends in neurosciences*, 22(4):146–51, April 1999a. 179

P Redgrave, TJ Prescott, and K Gurney. The basal ganglia: a vertebrate solution to the selection problem? *Neuroscience*, 89:1009–1023, 1999b. 171

Peter Redgrave, Kevin Gurney, and John Reynolds. What is reinforced by phasic dopamine signals? *Brain research reviews*, 58(2):322–39, August 2008. 179

Jon Reed, Robert Toombs, and Nils Aall Barricelli. Simulation of Biological Evolution and Machine Learning. *Journal of theoretical biology*, 17:319–342, 1967. 30

Will Regan, Floris Van Breugel, and Hod Lipson. Towards Evolvable Hovering Flight on a Physical Ornithopter. In Luis Mateus Rocha, Larry S. Yaeger, Mark A. Bedau, Dario Floreano, Robert L. Goldstone, and Alessandro Vespignani, editors, *Artificial Life X, Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 241–247. The MIT Press, 2002. 78

T. Reil and P. Husbands. Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, 6:159–168, 2002. 102, 139

Ricardo a M Reis, Ana Lucia M Ventura, Regina C C Kubrusly, Maria Christina F de Mello, and Fernando G de Mello. Dopaminergic signaling in the developing retina. *Brain research reviews*, 54(1):181–8, April 2007. 180

R.A. Rescorla and A.R. Wagner. *Classical Conditioning II: Current Research and Theory*, chapter A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement, pages 64–99. Appleton Century Crofts, New York, 1972. 186

Jeremy R Reynolds and Randall C O'Reilly. Developing PFC representations using reinforcement learning. *Cognition*, 113(3):281–92, December 2009. 182

Marios Richards. Does learning accelerate evolution? *Work*, 2008. 34

T. Riel. Artificial Evolution of Neural Controllers in a Real-Time PhysicsEnvironment. Master's thesis, Evolutionary and Adptive Systems, COGS, University of Sussex, Brghton, UK, 1999. 112

S. Risi, C. E. Hughes, and K. O. Stanley. Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 18(6):470–491, October 2010. 38

Michael G. Ritchie. Sexual Selection and Speciation. *Annual Review of Ecology, Evolution, and Systematics*, 38(1):79–102, December 2007. 37

Patrick D Roberts and Curtis C Bell. Spike timing dependent synaptic plasticity in biological systems. *Biological cybernetics*, 87(5-6):392–403, December 2002. 240

Patrick D Roberts, Roberto a Santiago, and Gerardo Lafferriere. An implemen-

tation of reinforcement learning based on spike timing dependent plasticity. *Biological cybernetics*, 99(6):517–23, December 2008. 240

Raul Rojas. *Neural networks: a systematic introduction*. Springer-Verlag, Berlin, 1996. 38

R Rosen. *Anticipatory Systems: Philosophical, Mathematical and Methodological Foundations*. Pergamon Press, 1985. 213

F Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, November 1958a. 42

Frank Rosenblatt. The perceptron: A theory of statistical separability in cognitive system. Technical Report No. VG-1196-G-1, Cornell Aeronautical Laboratory, Inc, Buffalo, USA, 1958b. 41

P M Rowe. Neural networks: a bridge between neuroscience and psychology. *Molecular medicine today*, 1(4):168–73, July 1995. 39

David E Rumelhart and James L McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, 1986. 38, 43

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 1986. 43

Eytan Ruppin. Evolutionary autonomous agents: a neuroscience perspective. *Nature reviews. Neuroscience*, 3(2):132–41, February 2002. 70

A.G. Rust, R. Adams, S. George, and H. Bolouri. Towards ComputationalNeuralSystemsThroughDevelopmentalSystems. In S. Wermter, J. Austin, and D. Willshaw, editors, *EmergentNeuralComputationalArchitecturesBased on Neuroscience*. Springer-Verlag, Berlin, 2001. 38

L Sagan. On the origin of mitosing cells. *Journal of Theoretical Biology*, 14(3): 255–274, 1967. 27

John D Salamone. Functions of mesolimbic dopamine: changing concepts and shifting paradigms. *Psychopharmacology*, 191(3):389, April 2007. 179

R D Samson, M J Frank, and Jean-Marc Fellous. Computational models of reinforcement learning: the role of dopamine as a reward signal. *Cognitive neurodynamics*, 4(2):91–105, June 2010. 181, 210

José Santos and Ángel Campo. Biped locomotion control with evolved adaptive center-crossing continuous time recurrent neural networks. *Neurocomputing*, 86:86–96, February 2012. 141

B. Sareni and L. Krahenbuhl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106, 1998. 81

Wim Hm Saris, Steven B Heymsfield, and William J Evans. Human brain evolution: food for thoughts. *Current opinion in clinical nutrition and metabolic care*, 11(6):683–5, November 2008. 221

Gregory M Saunders and Jordan B Pollack. The Evolution of Communication Schemes Over Continuous Channels. In Pattie Maes, Maja J. Mataric, Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson, editors, *From Animals*

to Animats 4 Proceedings of the Fourth International Conference on Adaptive Behaviour, (SAB 1996), Cape Cod, Massachusetts, 1996. The MIT Press. 77

M.W. Sayers. *Symbolic Computer Methods to Automatically Formulate Vehicle SimulationCodes*. PhD thesis, The University of Michigan, 1990. 112

Daniel L Schacter, Gagan S Wig, and W Dale Stevens. Reductions in cortical activity during priming. *Current opinion in neurobiology*, 17(2):171–6, April 2007. 220

J.D. Schaffer, D. Whitley, and L.J. Eshelman. Combinations of genetic algorithms and neural networks: a survey of the state of the art. In *[Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pages 1–37. IEEE Comput. Soc. Press, 1992. 66

J.David Schaffer, Richard a. Caruana, and Larry J. Eshelman. Using genetic search to exploit the emergent behavior of neural networks. *Physica D: Nonlinear Phenomena*, 42(1-3):244–248, June 1990. 66

Massimiliano Schembri, Marco Mirolli, and Gianluca Baldassarre. Evolving internal reinforcers for an intrinsically motivated reinforcement-learning robot. *2007 IEEE 6th International Conference on Development and Learning*, pages 282–287, July 2007. 292

Matthew Schlesinger and Domenico Parisi. Connectionism in an Artificial Life Perspective: Simulating Motor, Cognitive, and Language Development. In D. Mareschal, S. Sirois, and G. Westermann, editors, *Neuroconstructivism* -

II: Perspectives and Prospects (Developmental Cognitive Neuroscience), pages 129–158. Oxford University Press, Oxford, UK, 2007. 74

N a Schmajuk and J J DiCarlo. A neural network approach to hippocampal function in classical conditioning. *Behavioral neuroscience*, 105(1):82–110, February 1991. 186

Jacob Schrum and Risto Miikkulainen. Evolving Multi-modal Behavior in NPCs. In *Proceedings of the 5th international conference on Computational Intelligence and Games*, pages 325–332. IEEE Press, 2009. 80

W Schultz. Adaptive dopaminergic neurons report the appetitive value of environmental stimuli. In J.W. Donahoe Dorsel and V. Packard, editors, *Neural-Networks Models of Cognition*. 1997. 186

W. Schultz. Predictive reward signal of dopamine neurons. *Journal of neurophysiology*, 80(1):1, 1998. xvi, xxiii, xxxi, 170, 222, 223, 275

W. Schultz, R. Romo, T. Ljungberg, J. Mirenowicz, J.R. Hollerman, and A. Dickinson. Reward-related Signals Carried by Dopamine Neurons. In James C. Houk, Joel L. Davis, and David G. Beiser, editors, *Models of Information Processing in the Basal Ganglia (Computational Neuroscience)*, chapter 12, page 233. The MIT Press, 1994. xvi, xxxii, 295, 296

W Schultz, L Tremblay, and J R Hollerman. Reward prediction in primate basal ganglia and frontal cortex. *Neuropharmacology*, 37(4-5):421–9, 1998. 178, 180

W. Schultz, L. Tremblay, and J.R. Hollerman. Reward processing in primate orbitofrontal cortex and basal ganglia. *Cerebral. Cortex*, 10:272284, 2000. 242

Wolfram Schultz. Introduction. Neuroeconomics: the promise and the profit. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 363(1511):3767–9, December 2008. 180

Wolfram Schultz. Dopamine signals for reward value and risk: basic and recent data. *Behavioral and brain functions : BBF*, 6:24, January 2010. 179

Wolfram Schultz, Kerstin Preuschoff, Colin Camerer, Ming Hsu, Christopher D Fiorillo, Philippe N Tobler, and Peter Bossaerts. Explicit neural signals reflecting reward uncertainty. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 363(1511):3801–11, December 2008. 179

Jacob T Schwartz. The New Connectionism : Developing Relationships and Neuroscience. *Daedalus*, 117(1):123–141, 1988. 39

Frederick Seitz. Language and Machines: Computers in Translation and Linguistics. Technical report, A Report Committee, Automatic Language Processing Advisory Sciences, Division of Behavioral Sciences, National Academy of Council, National Research, 1966. 50

Chad W Seys and Randall D Beer. Evolving Walking: The Anatomy of an Evolutionary Search. In S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, and J-A. Meyer, editors, *From Animals to Animats 8: Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior*, pages 357–363. MIT Press, 2004. 76

Chad W. Seys and Randall D. Beer. Effect of Encoding on the Evolvability of an Embodied Neural Network Categories and Subject Descriptors. In *GECCO*

2006 Workshop Proceedings, Workshop on Complexity through Development and Self-Organizing Representations (CODESOAR), 2006. 76

Chad W Seys and Randall D Beer. Genotype Reuse More Important than Genotype Size in Evolvability of Embodied Neural Networks. In Fernando Almeida e Costa, Luis Mateus Rocha, Ernesto Costa, Inman Harvey, and António Coutinho, editors, *Advances in Artificial Life: 9th European Conference, ECAL 2007*, pages 915–924. Springer, 2007. 76

C E Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(July 1928):379–423, 1948. 44

Yoonsik Shim and Phil Husbands. Chaotic Exploration and Learning of Locomotion Behaviors. *Neural Computation*, 24:1–38, 2012. 73, 141

Shy Shoham, Daniel H O'Connor, and Ronen Segev. How silent is the brain: is there a "dark matter" problem in neuroscience? *Journal of comparative physiology. A, Neuroethology, sensory, neural, and behavioral physiology*, 192 (8):777–84, August 2006. 220

Daphna Shohamy. Learning and motivation in the human striatum. *Current opinion in neurobiology*, 21(3):408–414, June 2011. 179

Laurent Siklssy. On the Evolution of Artificial Intelligence. *Information Sciences*, 2(4):369–377, 1970. 50

Herbert A Simon. Artificial Intelligence Artificial intelligence: an empirical science. *Artificial Intelligence*, 77:95–127, 1995. 52

Karl Sims. Artificial evolution for computer graphics. *Computer Graphics*, 25(4): 319–328, 1991. 68

Karl Sims. Evolving Virtual Creatures. In *SIGGRAPH '94 Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, number July, pages 15–22, 1994. 68

John Maynard Smith and George Robert Price. The logic of animal conflict. *Nature*, 246(15-18), 1973. 26

T. Smith, P. Husbands, and M. O'Shea. Not measuring evolvability: initial investigation of an evolutionary robotics search space. *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 1:9–16, 2001a. 72

T. Smith, A. Philippides, P. Husbands, and M. O'Shea. Neutrality and ruggedness in robot landscapes. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, 2:1348–1353, 2002a. 72

T. Smith, P. Husbands, A. Philippedes, and M. O'Shea. TemporallyAdaptiveNetworks:Analysis of GasNet Robot Controllers. In R. Standish, M.A. Bedau, and H. A. Abbass, editors, *Artificial Life VIII: Proceedings of the Eight International Conference on Artificial Life*, pages 274–282. MIT Press, 2003a. 107, 139

T.M.C. Smith, P. Husbands, A. Philippides, and M. O'Shea. M. neuronal plasticity and temporal adaptivity: Gasnet robot control networks. *Adaptive Behavior*, 10(3/4):161–184, 2002b. 136

T.M.C. Smith, P. Husbands, A. Philippides, and M O'Shea. Neuronal Plasticity and Temporal Adaptivity : GasNet Robot Control Networks. *Adaptive Behavior*, 10(3-4):161–184, 2002c. 72

Tom Smith, Phil Husbands, and Michael O Shea. Neutral Networks and Evolvability with Complex Genotype-Phenotype Mapping. In Jozef Kelemen and Petr Sosik, editors, *Advances in Artificial Life: 6th European Conference, ECAL 2001, Proceedings*, pages 272–281. Springer, 2001b. 72

Tom Smith, Phil Husbands, Paul Layzell, and Michael O'Shea. Fitness landscapes and evolvability. *Evolutionary computation*, 10(1):1–34, January 2002d. 72

Tom Smith, Phil Husbands, Andy Philippides, and Michael O Shea. Temporally adaptive networks : Analysis of GasNet robot control networks. In Russell Standish, Hussein A. Abbass, and Mark A. Bedau, editors, *Artificial Life VIII: Proceedings of the Eighth International Conference on Artificial Life (Complex Adaptive Systems)*, pages 274–282. A Bradford Book, 2002e. 72, 104

Tom Smith, Phil Husbands, and Michael O'Shea. Local Evolvability of Statistically Neutral HasNet Robot Controllers. *Biosystems*, 69:223–243, 2003b. 72

Yoland Smith, Dinesh V Raju, Jean-Francois Pare, and Mamadou Sidibe. The thalamostriatal system: a highly specific network of the basal ganglia circuitry. *Trends in neurosciences*, 27(9):520–7, September 2004. 300

P Smolensky. On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11(1):1–74, 1988. 53, 54

Andrea Soltoggio, P. Durr, Claudio Mattiussi, and Dario Floreano. Evolving

neuromodulatory topologies for reinforcement learning-like problems. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 2471–2478. IEEE, 2008. 292

Valerio Sperati, Vito Trianni, and Stefano Nolfi. Self-organised path formation in a swarm of robots. *Swarm Intelligence*, 5(2):97–119, April 2011. 74

Kenneth O. Stanley. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*, 10(2):99–127, 2002a. 80

Kenneth O Stanley. Efficient Reinforcement Learning through Evolving Neural Network Topologies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, San Francisco, CA, 2002b. Morgan Kaufmann. 80

Kenneth O. Stanley. *Genetic Programming and Evolvable Machines*, pages 1–36, 2007. 81

Kenneth O Stanley and Risto Miikkulainen. A Taxonomy for Artificial Embryogeny. *Artificial Life*, 9(2):93–130, 2003. 81

Kenneth O Stanley, David D Ambrosio, and Jason Gauci. A Hypercube-Based Indirect Encoding for Evolving Large-Scale Neural Networks. *Artificial Life*, 15(2):185–212, 2009. 81, 82

Luc Steels. Evolving grounded communication for robots. *Trends in Cognitive Sciences*, 7(7):308–312, July 2003. 29

Marcus Stephenson-Jones, Ebba Samuelsson, Jesper Ericsson, Brita Robertson, and Sten Grillner. Evolutionary conservation of the Basal Ganglia as a common

vertebrate mechanism for action selection. *Current biology : CB*, 21(13):1081–91, July 2011. 171

Bryan a Strange, Andrew Duggins, William Penny, Raymond J Dolan, and Karl J Friston. Information theory, novelty and hippocampal responses: unpredicted or unpredictable? *Neural networks : the official journal of the International Neural Network Society*, 18(3):225–30, April 2005. 227

G.F. Striedter. *Principles of Brain Evolution*. Sinauer Asscoiates, Inc, 2005. 269, 300

R.E. Suri and W. Schultz. Temporal difference model reproduces anticipatory neural activity. *Neural Computation*, 13(4):841–862, 2001. 186

D.J. Surmeier, J Eberwine, C.J. Wilson, Y. Cao, A. Stefani, and S.T. Kitai. Dopamine recceptor subtypes colocalize in rat striatonigral neurons. *Proceedings of the National Academy of Sciences of the U.S.A.*, 89:10178–10182, 1992. 295

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998. 170, 178, 210, 211, 212

Maki Suzuki and JD Johnson. Decrements in Hippocampal Activity with Item Repetition during Continuous Recognition: An fMRI Study. *Journal of Cognitive Neuroscience*, pages 1–11, 2008. 220

Can Ozan Tan and Daniel Bullock. A local circuit model of learned striatal and dopamine cell responses under probabilistic schedules of reward. *The Journal*

of neuroscience : the official journal of the Society for Neuroscience, 28(40): 10062–74, October 2008. 179

A.I. Tauber. Reframing developmental biology and building evolutionary theory's new synthesis. *Perspectives in biology and medicine*, 53(2):257, 2010. 28

Matthew E. Taylor, Shimon Whiteson, and Peter Stone. Comparing evolutionary and temporal difference methods in a reinforcement learning domain. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, number July, pages 1321–1328, 2006. 292

X Thibert-Plante and a P Hendry. Factors influencing progress toward sympatric speciation. *Journal of evolutionary biology*, 24(10):2186–96, October 2011. 37

Adrian Thompson. An evolved circuit, intrinsic in silicon, entwined with physics. In T. Higuchi, M. Iwata, and L Weixin, editors, *Proceedings of the 1st International Conference on Evolvable Systems (ICES'96)*, pages 406–422. Springer-Verlag, 1996. 73

Adrian Thompson. On the automatic design of robust electronics through artificial evolution. In M. Sipper, D. Mange, and E Perez-Uribe, editors, *Proceedings of the 2nd International Conference on Evolvable Systems (ICES'98)*, pages 13–24. Springer-Verlag, 1998. 73

Evan Thompson and Francisco J. Varela. Radical embodiment: neural dynamics and consciousness. *Trends in cognitive sciences*, 5(10):418–425, October 2001. 62

Vito Trianni and Stefano Nolfi. Engineering the evolution of self-organizing behaviors in swarm robotics: a case study. *Artificial life*, 17(3):183–202, January 2011. 74

RL Trivers. The evolution of reciprocal altruism. *Quarterly review of biology*, 46 (1):35–57, 1971. 26

Leonardo Trujillo, Gustavo Olague, E. Lutton, F. Fernández de Vega, L. Dozal, and Eddie Clemente. Speciation in Behavioral Space for Evolutionary Robotics. *Journal of Intelligent & Robotic Systems*, pages 1–29, 2011. 38, 83

E. Tuci and M. Quinn. Behavioural Plasticity in Autonomous Agents: A Comparison Between Two Types of Controller. In S. Cagnoni, J.J.R. Cardalda, D.W. Corne, J. Gottlieb, A. Guillot, E. Hart, C.G. Johnson, E. Marchiori, J.-A. Meyerand M. Middendorf, and G.R. Raidl, editors, *Applications of Evolutionary Computing: EvoWorkshops 2003*, pages 661–672, Berlin, 2003. Springer-Verlag. 119

Elio Tuci, Tomassino Ferrauto, Arne Zeschel, Gianluca Massera, and Stefano Nolfi. An Experiment on Behavior Generalization and the Emergence of Linguistic Compositionality in Evolving Robots. *IEEE Transactions on Autonomous Mental Development*, 3(2):176–189, June 2011. 74

A M Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950. 47

Alan M. Turing. *The Essential Turing*. Oxford University Press, USA, 2004. 46

Alan Turring. Turing letter to W. Ross Ashby, 1946. 47

Stuart a. Umpleby. Ross Ashby's general theory of adaptive systems. *International Journal of General Systems*, 38(2):231–238, February 2009. 45

D. Floreano Urzelai. and J. Artificial Evolution of Adaptive Software: An Application to Autonomous Robots. *3D The Journal of Three dimensional images (in japanese)*, 14(4):64–69, 2000. 75

J. Urzelai and D. Floreano. Evolutionary Robots with Fast Adaptive Behavior in New Environments. In A. Thompson, editor, *Third International Conference on EvolvableSystems: From Biology to Hardware (ICES2000)*. Springer Verlag, 2000a. 109, 117

J Urzelai and D Floreano. Evolution of adaptive synapses: robots with fast adaptive behavior in new environments. *Evolutionary computation*, 9(4):495–524, January 2001. 75, 99, 101

J. Urzelai, D. Floreano, M. Dorigo, and M. Colombetti. Incremental Robot Shaping. *Connection Science*, 10:341–360, 1998. 75

Joseba Urzelai and Dario Floreano. Evolutionary robotics: coping with environmental change. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2000)*, 2000b. 35, 38

Elizabeth R Valentine. Theoretical Perspective - Neural Nets : From Hartley and Hebb to Hinton. *Journal of Mathematical Psychology*, 33:348–357, 1989. 39

F.J. Varela, E. Thompson, and E. Rosch. *The embodied mind: Cognitive science and human experience*. The MIT Press, 1992. 57

Francisco J Varela, Evan Thompson, and Eleanor Rosch. *The Embodied Mind: Cognitive Science and Human Experience*, volume 1. MIT Press, 1991. 60

Franciso J. Varela. The re-enchantment of the concrete. In Luc Steels and Rodney Brooks, editors, *The Artificial Life route to Artificial Intelligence: Building Embodied, Situated Agents.*, pages 320–338. Psychology Press, 1995. 58, 59

Patricia A Vargas, Ezequiel A Di Paolo, and Phil Husbands. A study of GasNet spatial embedding in a delayed-response task. In Seth Bullock, Jason Noble, Richard A. Watson, and Mark A. Bedau, editors, *Artifcial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 640–647. MIT Press, 2008. 72

Evan a. Variano and Hod Lipson. Networks, Dynamics, and Modularity. *Physical Review Letters*, 92(18):1–4, May 2004. 78

Eric D Vaughan, Ezequiel Di Paolo, and Inman R Harvey. The tango of a load balancing biped. In *Proceedings of CLAWAR 2004, 7th International Conference on Climbing and Walking Robots*, Madrid, Spain, 2004. 73

B Vincent. Synaptic energy efficiency in retinal processing. *Vision Research*, 43 (11):1285–1292, May 2003. 220

Benjamin T. Vincent, Roland J. Baddeley, Tom Troscianko, and Iain D. Gilchrist. Is the early visual system optimised to be energy efficient? *Network: Computation in Neural Systems*, 16(2-3):175–190, January 2005. 220

Julien Vitay and Fred H Hamker. A computational model of Basal Ganglia and

its role in memory retrieval in rewarded visual memory tasks. *Frontiers in computational neuroscience*, 4(May):1–18, January 2010. 181

Julien Vitay, Jérémy Fix, Frederik Beuth, Henning Schroll, and F.H. Hamker. Biological Models of Reinforcement Learning. *eyeshots.it*, pages 12–18, 2009. 170, 179

Heinz von Foesrter. Computation in Neural Nets. *Currents In Modern Biology*, 1:47–93, 1967. 43

J. von Neumann. First draft of a report on the EDVAC. *IEEE Annals of the History of Computing*, 15(4):27–75, 1993. 46

Scott Waddell. Dopamine reveals neural circuit mechanisms of fly memory. *Trends in neurosciences*, 33(10):457–464, August 2010. 171

C.H. Waddington. "evolutionary adaptation". *"Perspectives in Biology and Medicine"*, 2(4):379–401, 1959. 36

Günter P. Wagner. Homologues, Natural Kinds and the Evolution of Modularity. *Integrative and Comparative Biology*, 36(1):36–43, 1996. 36

Markus Waibel, Laurent Keller, and Dario Floreano. Genetic Team Composition and Level of Selection in the Evolution of Cooperation. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 13(3):648–660, 2009. 29

Alfred R Wallace. On the law which has regulated the introduction of new species. *Annals and Magazine of Natural History Series 2*, 16(93):184–196, 1855. 25

W Grey Walter. An imitation of life. *Scientific American*, 1950. 45

W Grey Walter. A machine that learns. *Scientific American*, (August), 1951. 45

C.J.C.H Watkins and P Dayan. Q-learning. *Machine Learning*, Volume 8:279–292, 1996. 212

J D Watson. *The Double Helix*, volume III. Atheneum, 1968. 25

Richard A Watson and Jordan B Pollack. Mutualism, Parasitism, and Evolutionary Adaptation. In Mark Bedau, John S. McCaskill, Norman H. Packard, Steen Rasmussen, John McCaskil, and Norman Packard, editors, *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life (Complex Adaptive Systems)*. A Bradford Book, 1992. 77

Richard A. Watson and Jordan B. Pollack. Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots. In *Proc. Congr. on Evolutionary Computation*, pages 335–342, Washington, DC, 1999. IEEE Press. 77

Richard A. Watson, Sevan G. Ficici, and Jordan B. Pollack. Embodied Evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18, April 2002. 78

Gerhard Weiss. Towards the synthesis of neural and evolutionary learning. *Progress in neural networks*, 5:1–33, 1993. 70

P J Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974. 43

Stuart a West, Ashleigh S Griffin, and Andy Gardner. Evolutionary explanations for cooperation. *Current biology : CB*, 17(16):R661–72, August 2007. 26

M. J. D. White. Models of Speciation. *American Association for the Advancement of Science*, 159(3819):1065–1070, 1968. 37

Paul White, Victor Zykov, Josh Bongard, and Hod Lipson. Three Dimensional Stochastic Reconfiguration of Modular Robots. In *Proceedings of Robotics Science and Systems*, Cambridge MA, 2005. MIT Press. 78

S. Whiteson, M. E. Taylor, and P. Stone. Empirical Studies in Action Selection with Reinforcement Learning. *Adaptive Behavior*, 15(1):33–50, March 2007. 292

D Whitley, T Starkweather, and C Bogart. Genetic algorithms and neural networks: optimizing connections and connectivity. *Parallel Computing*, 14(3): 347–361, August 1990. 66

Darrell Whitley, Stephen Dominic, and Rajarshi Das. Genetic reinforcement learning for neurocontrol problems. *Machine Learning*, 284:259–284, 1993. 292

L.D. Whitley and T Hanson. Optimizing neural networks using faster more accurate genetic search. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 391–397, San Francisco, CA, 1989. Morgan Kaufmann Publishers Inc. 66

T Wichmann and J O Dostrovsky. Pathological basal ganglia activity in movement disorders. *Neuroscience*, June 2011. 172

N Wiener. Cybernetics. *Scientific American*, (179):14–19, 1948. 44

Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*, volume 11. John Wiley & Sons, New York, 1949. 44

G C Williams. *Adaptation and Natural Selection*, volume 13. Princeton University Press, 1966. 25

Paul L Williams, Randall D Beer, and Michael Gasser. An Embodied Dynamical Approach to Relational Categorization. In B.C. Love, K. McRae, and V.M. Sloutsky, editors, *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, volume 1, pages 223–228. Psychology Press, 2008. 76, 77

D. Willshaw, D. Dennett, and D. Partridge. Non-Symbolic Approaches to Artificial Intelligence and the Mind [and Discussion]. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 349 (1689):87–102, October 1994. 52

C.J. Wilson. *The Synaptic Organization of the Brain*, chapter 9, pages 361–413. Oxford University Press, 5th edition, 2004. xvi, xviii, xxxiii, 172, 174, 305

C.J Wilson and Y Kawaguchi. The Origins of Two-State Spontaneous Membrane Fluctuations of Neostriatal Spiny Neurons. *The Journal of Neuroscience*, 16 (7):2397–2410, 1996. 294

Margaret Wilson. Six views of embodied cognition. *Psychonomic bulletin & review*, 9(4):625–36, December 2002. 57

BC Wittmann, ND Daw, and Ben Seymour. Striatal activity underlies novelty-based choice in humans. *Neuron*, 58(6):967–973, 2008. 197

Florentin Wörgötter and Bernd Porr. Temporal sequence learning, prediction, and control: a review of different models and their relation to biological mechanisms. *Neural computation*, 17(2):245–319, February 2005. 187

Sewall Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. *Proceedings of the sixth international congress on Genetics*, pages 355–366, 1932. 33

V C Wynne-Edwards. Animal dispersion in relation to social behaviour. *Canadian Medical Association Journal*, 88(25):1255–1256, 1963. 26

J Z Xiang and M W Brown. Differential neuronal encoding of novelty, familiarity and recency in regions of the anterior temporal lobe. *Neuropharmacology*, 37 (4-5):657–76, 1998. 197

B. Yamauchi and R.D. Beer. Integrating Reactive, Sequential and Learning Behavior Using Dynamical Neural Networks. In D. Cliff, P. Husbands, J. Meyer, and S. Wilson, editors, *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB 94)*, pages 205–224. MIT Press, 1994a. 128

B. M. Yamauchi and R. D. Beer. Sequential Behavior and Learning in Evolved Dynamical Neural Networks. *Adaptive Behavior*, 2(3):219–246, January 1994b. 76

Brian Yamauchi and Randall D. Beer. Integrating Reactive, Sequential, and Learning Behavior Using Dynamical Neural Networks. In D. Cliff, P. Husbands, J. Meyer, and S. Wilson, editors, *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB 94)*, pages 382–391, 1994c. 76, 292

Holly Yanco and Lynn Andrea Stein. An Adaptive Communication Protocol for Cooperating Mobile Robots. In Jean-arcady Meyer, editor, *From Animals to*

Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior (Complex Adaptive Systems), pages 478–486. MIT Press, 1993. 29

X Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999. 70

Xin Yao. A Review of Evolutionary Artificial Neural Networks. *International Journal of Intelligent Systems*, 8:539–567, 1993. 66

Jason Yosinski, Jeff Clune, Diana Hidalgo, Sarah Nguyen, Juan Cristobal Zagal, and Hod Lipson. Evolving Robot Gaits in Hardware : the HyperNEAT Generative Encoding Vs . Parameter Optimization. In Tom Lenaerts, Mario Giacobini, Hugues Bersini, Paul Bourgine, Marco Dorigo, and René Doursat, editors, *Advances in Artifcial Life, ECAL 2011 (Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems)*, pages 890–897. MIT Press, 2011. 82

Hongnian Zang, Shujun Zhang, and Kevin Hapeshi. A Review of Nature-Inspired Algorithms. *Journal of Bionic Engineering*, 7:S232–S237, September 2010. 33

Tom Ziemke. Whats that Thing Called Embodiment? In Alterman and Kirsh, editors, *Proceedings of the 25th Annual Conference of the Cognitive Science Society*, pages 1134–1139, Mahwah, NJ, 2005. Lawrence Erlbaum. 57

Jean-Christophe Zufferey, Alexis Guanella, Antoine Beyeler, and Dario Floreano. Flying over the Reality Gap : From Simulated to Real Indoor Airships. *Autonomous Robots*, 21(3):243–254, 2006. 75

Victor Zykov, Efstathios Mytilinaios, Mark Desnoyer, and Hod Lipson. Evolved and Designed Self-Reproducing Modular Robotics. *IEEE Transactions on Robotics*, 23(2):308–319, 2007. 78